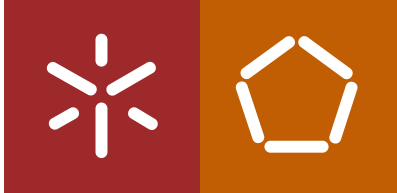


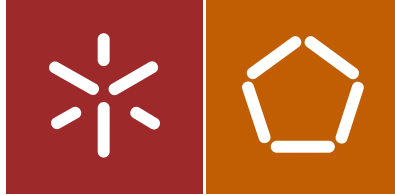


João Pedro Martins Figueiras

Gestão em linha de um armazém de
produtos alimentares através de um
sistema dinâmico de páginas de internet

Universidade do Minho
Escola de Engenharia





Universidade do Minho
Escola de Engenharia

João Pedro Martins Figueiras

Gestão em linha de um armazém de
produtos alimentares através de um
sistema dinâmico de páginas de internet

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao Grau de
Mestre em Engenharia Eletrónica Industrial
e Computadores

Trabalho efetuado sob a orientação do
Professor Doutor Agostinho Gil Lopes

Agradecimentos

A conclusão deste ciclo da minha vida não seria possível sem a ajuda de pessoas muito importantes para mim.

Um obrigado muito especial à Sónia Antunes, companheira, amiga e mais que tudo, por todo o apoio e incentivo.

Agradecer também ao Professor Doutor Agostinho Gil Lopes pelo apoio e ajuda enquanto orientador deste trabalho académico.

Gostaria de agradecer também aos meus amigos e família por estarem presentes e por permitirem que este ciclo seja concluído.

Resumo

Os armazéns de produtos alimentares, devido à natureza dos produtos que possuem, necessitam de cuidados especiais. Nomeadamente, devido às condições de temperatura nos locais em que determinados produtos são colocados, é necessário manter essas zonas do armazém dentro de determinados parâmetros de temperatura. O *Hazard Analysis and Critical Control Point* ou HACCP prevê que em determinadas zonas de armazenamento, como as arcas frigoríficas, seja feita a monitorização da temperatura pelo menos duas vezes ao dia.

Hoje em dia, devido à facilidade de acesso à internet, é possível fazer a monitorização destes parâmetros através de páginas de internet, mais concretamente através de uma aplicação *web*. Os dados são recolhidos por sensores colocados no armazém e enviados de forma automática para o sistema de monitorização. A aplicação *web* regista os valores e envia alertas por *email* quando existem desvios nas temperaturas pré-definidas. Este processo automático evita a necessidade de um deslocamento ao local para verificar se a temperatura se mantém dentro dos valores pré-definidos.

Este sistema de gestão, ou aplicação *web*, que faz a monitorização da temperatura é também uma ferramenta de trabalho para a empresa. Permite fazer gestão de *stock*, consultar tabelas de preços, consultar contas corrente e fichas de cliente, entre outros.

Palavras-Chave: Páginas da Internet, aplicação *web*, monitorização, sensores, HACCP;

Abstract

The warehouses of food products due to the nature of the products they have, need special care. Namely, because the temperature conditions in places where certain products are placed, it is necessary to keep these storage zones within certain parameters of temperature. The HACCP provides that in certain storage areas, such as freezers, be made a temperature monitoring at least twice a day.

Today, due to ease of access to the internet, it is possible to monitor these parameters through web pages, specifically through a web application. The data collected by sensors placed on the store are sent automatically to the monitoring system. The web application records the values and sends email alerts when there are deviations from the pre-set temperatures. This automated process avoids the need for a displacement to verify that the temperature is maintained within the pre-set values.

This management system or web application, which is a temperature monitoring, is also a working tool for the company. Allows to stock management, consulting price lists, refer to current accounts and customer records, among others.

Keywords: Web pages, web application, monitoring, sensors, HCCP;

"Life is like riding a bicycle. To keep your balance, you must keep moving." (Albert Einstein)

Conteúdo

1	Introdução	1
1.1	Contextualização e Motivação	2
1.2	Objetivos	3
1.3	Estrutura da Dissertação	4
2	Trabalhos Relacionados	5
2.1	Introdução	5
2.2	Estado da Arte	5
2.2.1	Aplicações dos sistemas de monitorização de dados	5
2.2.2	Aplicações da <i>Raspberry Pi</i>	6
2.2.3	<i>Raspberry Pi</i> como nó sensor	7
2.2.4	Redes de sensores	10
3	Fundamentos Teóricos	15
3.1	UML	15
3.2	Aplicações Nativas e Aplicações <i>Web</i>	15
3.3	Internet	16
3.4	Linguagens de Programação <i>Web</i>	17
3.4.1	HTML5 & CSS3	17
3.4.2	<i>JavaScript</i>	18
3.4.3	PHP e <i>MySQL</i>	19
3.5	Segurança nas Aplicações <i>Web</i>	20
3.5.1	Criptografia	20
3.5.2	Vulnerabilidades <i>Web</i>	21
3.6	<i>Hardware</i>	22
3.6.1	<i>Raspberry Pi</i>	22
3.6.2	Módulo <i>Wifi</i> para <i>Raspberry Pi</i>	25
3.6.3	DHT22	25
3.6.4	DS18B20	26

3.6.5	<i>Raspberry Pi</i> LCD 1602 <i>Add-on</i>	27
4	Metodologias	29
4.1	Visão Geral do Sistema	29
4.2	<i>Hardware</i>	29
4.3	<i>Software</i>	31
4.3.1	<i>Raspberry Pi</i>	31
4.3.2	Aplicação <i>web</i>	33
5	Resultados e Discussão	45
5.1	<i>Hardware</i>	45
5.2	Aplicação <i>Web</i>	46
5.2.1	Compatibilidade da aplicação <i>web</i>	46
5.2.2	Segurança	48
5.2.3	Funcionalidades implementadas na aplicação <i>web</i>	50
5.3	Gráficos de Temperatura	55
6	Conclusão e Trabalho Futuro	61
	Bibliografia	63

Lista de Figuras

2.1	Nó sensor [1].	7
2.2	Diagrama de blocos [2].	11
2.3	Diagrama de blocos [3].	12
2.4	Diagrama de blocos [4].	13
2.5	Diagrama de blocos [5].	14
2.6	Diagrama de blocos [6].	14
3.1	Diagrama <i>web</i> [7].	18
3.2	<i>Raspberry Pi</i> modelo b [1].	22
3.3	<i>Raspberry Pi</i> - Componentes [8].	23
3.4	Pinos GPIO [1].	24
3.5	Módulo wifi RTL8188CU [9].	25
3.6	Sensor DHT22 e dimensões em milímetros [10].	26
3.7	Sensor DS18B20	27
3.8	<i>Raspberry Pi</i> LCD 1602 <i>Add-on</i> [11].	28
4.1	Diagrama geral do sistema.	30
4.2	Módulo LCD1602 acoplado na <i>Raspberry Pi</i>	30
4.3	Esquemático elétrico de ligação do <i>hardware</i>	31
4.4	Fluxograma do programa executado na <i>Raspberry Pi</i>	32
4.6	Diagrama <i>Use Case</i>	37
4.7	Diagrama de estados da aplicação <i>web</i>	38
4.8	Diagrama da aplicação <i>web</i>	40
4.9	Diagrama EER.	41
4.5	Fluxograma das tarefas principais.	43
5.1	Montagem final do <i>hardware</i>	45
5.2	Página “Artigos” acedida de um computador.	47
5.3	Página “Artigos” acedida de um <i>tablet</i> de dez polegadas.	48
5.4	Redimensionamento da página “Artigos” em diversos dispositivos.	49

5.5	Página “ <i>Login</i> ” da aplicação <i>web</i>	52
5.6	Página “Registrar”.	52
5.7	Página “Utilizador” acedida num dispositivo de quatro polegadas. . .	53
5.8	Página “Todas as Contas Correntes” consultada num <i>tablet</i> de sete polegadas.	54
5.9	Página “Conta Corrente do Cliente”.	55
5.10	Documento PDF criado.	55
5.11	Página “Famílias” acedida num <i>tablet</i> de sete polegadas.	56
5.12	Página “Informação do Cliente” acedida num <i>tablet</i> de sete polegadas.	57
5.13	Gráficos de temperatura.	58
5.14	Página “Atualizar” acedida num <i>tablet</i> de sete polegadas.	58
5.15	Gráfico da arca 1.	59
5.16	Gráfico da arca 2.	59

Lista de Tabelas

2.1	Comparação tamanho, peso, custo[1]	8
2.2	Comparação processador, RAM, memória externa [1].	8
2.3	Comparação sistemas operativos [1].	10
3.1	Vulnerabilidades nas aplicações <i>web</i> [12].	21
3.2	Características <i>Raspberry Pi</i> [13].	24
3.3	Características sensor DHT22.	26
3.4	Características sensor DS18B20 [14].	28
5.1	Vulnerabilidades <i>web</i> .	50

Lista de Listagens

5.1	Criação da chave de segurança.	48
5.2	Função de <i>Hash</i>	49
5.3	Criação do <i>Salt</i>	49
5.4	Algoritmo <i>Blowfish</i>	49

Nomenclatura

ADC	Analog-to-Digital Converter
API	Application Programming Interface
APPCC	Análise de Perigos e Pontos Críticos de Controlo
ARM	Acorn RISC Machine
CO ₂	Dióxido de carbono
CSI	Camera Serial Interface
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
EER	Extended ER
ER	Entidade Relação
GPS	Global Positioning System
HCCP	Hazard Analysis and Critical Control Point
HDMI	High-Definition Multimedia Interface
HR	Humidade Relativa
HTML	HyperText Markup Language
I ² C	Inter-Integrated Circuit
IE9	Internet Explorer 9
IVA	Imposto Valor Acrescentado
LAN	Local Area Network

LCD	Liquid Crystal Display
MAC	Macintosh Operating System
OWASP	About The Open Web Application Security Project
PDF	Portable Document Format
PHP	Hypertext Preprocessor
RAM	Random Access Memory
SPI	Serial Peripheral Interface
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
USB	Universal Serial Bus
WSN	Wireless Sensor Network

1 Introdução

Hoje em dia, a segurança alimentar dos produtos comercializados é essencial para os consumidores. A evolução da indústria alimentar, em que novos métodos de produção e distribuição foram desenvolvidos, associados a novos hábitos alimentares, conduziu ao desenvolvimento de perigos que põem em causa a saúde dos consumidores. Posto isto, é essencial um controlo eficaz da higiene e segurança dos alimentos em toda a cadeia de fabrico, armazenamento e distribuição.

Com o propósito de controlar a segurança e higiene dos alimentos foram desenvolvidas técnicas de controlo da segurança alimentar. O HACCP, também conhecido como Análise de Perigos e Pontos Críticos de Controlo (APPCC) é um sistema de gestão de segurança alimentar que tem como base a análise de todas as etapas de produção dos alimentos. É feita uma análise cuidada dos pontos críticos no processamento dos alimentos e são implementados meios de controlo nesses mesmos pontos [15]. Um dos pontos críticos no armazenamento é a temperatura a que os produtos estão expostos. Os produtos mais sensíveis à variação da temperatura, possuem uma temperatura a que devem ser armazenados.

O intervalo de temperatura entre os 4 ° e os 63 °C é denominado de zona de perigo, uma vez que é propícia a multiplicação de microorganismos. Para garantir que esta gama de valores não é atingida implementa-se um meio de monitorização deste ponto crítico de controlo. O HACCP impõe uma monitorização mínima de duas vezes ao dia da temperatura das arcas frigoríficas instaladas, uma na parte da manhã e outra na parte da tarde [15]. Através desta monitorização consegue-se perceber se ocorreram variações nos valores da temperatura que ponham em causa a conservação dos alimentos.

Nos dias que correm, a tecnologia assume uma papel fundamental para o desenvolvimento de novos métodos de monitorização. A evolução dos dispositivos móveis e a facilidade com que hoje em dia se acede à internet, permite a criação de sistemas de monitorização mais funcionais. Estes sistemas possuem a característica de poderem ser acedidos em qualquer momento e onde quer que estejamos, partindo

sempre do pressuposto que existe uma ligação à internet.

As aplicações *web*, cada vez mais desenvolvidas, acedidas através dos dispositivos móveis, são uma ferramenta útil e podem ser usadas para implementar sistemas de monitorização. No entanto, devido à evolução das linguagens de programação *web*, as aplicações podem também possuir muitas outras ferramentas úteis que podem ser usadas nos processos das empresas.

1.1 Contextualização e Motivação

Esta dissertação foi desenvolvida no contexto de uma empresa que comercializa produtos alimentares. Esta empresa possui duas arcas frigoríficas para armazenamento de produtos sensíveis à exposição a temperaturas elevadas. É relevante referir que apenas uma das arcas possui produtos que tem de estar numa gama de temperatura que pode variar entre 1 ° e 4 °C. A outra arca frigorífica tem o propósito de armazenar produtos com uma gama de valores que podem atingir no máximo os 20 °C.

A monitorização destes dois pontos críticos é feita duas vezes ao dia por um funcionário da empresa, é necessário preencher um documento com os valores da temperatura das arcas e a hora a que essa monitorização foi efetuada. Com o intuito de facilitar a monitorização e evitar a deslocação de uma pessoa duas vezes por dia para verificar os valores da temperatura, pretende-se com esta dissertação automatizar este processo de monitorização.

Com este propósito, foi desenvolvido um sistema de monitorização automático. Os valores da temperatura das duas câmaras são obtidos por sensores e registados numa base de dados através de uma *Raspberry Pi*. Estes valores podem ser depois consultados através de uma aplicação *web*, acedida através de um dispositivo com acesso à internet.

Por outro lado, a aplicação *web* é também uma ferramenta importante na gestão da empresa. Possui funcionalidades que permitem visualizar dados importantes para o utilizador. Sendo por isso, uma ferramenta de trabalho essencial que ajuda a simplificar os processos da empresa.

1.2 Objetivos

O objetivo desta dissertação é desenvolver uma solução de monitorização e de simplificação dos processos da empresa. Esta solução pode ser dividida em duas partes distintas:

- Monitorização dos valores de temperatura.
- Aplicação *web*.

Relativamente à monitorização dos valores de temperatura das arcas frigoríficas, pretende-se que cumpram os seguintes objetivos:

- Evitar a deslocação duas vezes por dia de uma pessoa para o registo dos valores de temperatura.
- Registar os valores automaticamente numa base de dados para posterior consulta através de uma aplicação *web*.
- Enviar alertas por *email* quando os valores da temperatura estão fora da gama de valores pré-definidos.

Quanto à aplicação *web*, deve através de um sistema dinâmico de páginas de internet, cumprir com os seguintes objetivos:

- Apenas permitir o acesso de utilizadores registados.
- Registo de novos utilizadores.
- Alterar *password* de acesso dos utilizadores.
- Visualização de gráficos de temperatura com os valores guardados na base de dados.
- Visualizar, guardar ou enviar por *email* os dados relativos às contas correntes dos clientes.
- Visualizar todos os documentos referentes à conta corrente dos clientes, presentes na base de dados, para consulta por valor do documento.
- Consultar dados dos clientes.

- Consultar informações sobre os produtos comercializados, incluindo o *stock* disponível de cada produto.
- Garantir a compatibilidade da aplicação com diversos *browsers*.
- Garantir a compatibilidade da aplicação com dispositivos com sistemas operativos diferentes.
- Garantir a compatibilidade da aplicação com dispositivos de dimensões de ecrã diferentes.

1.3 Estrutura da Dissertação

Este documento é constituído por vários capítulos, de forma a ser apresentado todo o processo de desenvolvimento da solução proposta.

Capítulo 2: Descreve todos os trabalhos analisados que direta ou indiretamente se relacionam com esta dissertação.

Capítulo 3: Descreve todos os fundamentos teóricos essenciais para a implementação da solução.

Capítulo 4: Apresenta as metodologias utilizadas com o objetivo de facilitar a implementação da solução.

Capítulo 5: Apresenta os resultados obtidos de acordo com os objetivos propostos inicialmente.

Capítulo 6: Para finalizar, este capítulo analisa todo o trabalho desenvolvido, bem como descreve o trabalho futuro.

2 Trabalhos Relacionados

Este capítulo tem como objetivo referenciar os vários projetos estudados no âmbito desta dissertação. Todos os trabalhos analisados estão direta ou indiretamente relacionados com projeto proposto nesta dissertação.

2.1 Introdução

A evolução dos dispositivos móveis, nomeadamente os *smartphones* e os *tablets*, em conjunto com a evolução das linguagens de programação e com a facilidade de acesso à internet, fez com que o desenvolvimento de aplicações *web* crescesse muito nos últimos anos. Muitos projetos nasceram baseados em aplicações *web*, sendo a área da monitorização de sensores através de aplicações *web* uma área em crescimento na atualidade. Com o objetivo de encontrar soluções que cumpram os objetivos desta dissertação, foi feito um levantamento de soluções para problemas idênticos aos problemas que se pretende resolver neste projeto.

Neste capítulo vão ser apresentadas soluções de sistemas de aquisição de dados com a intenção de encontrar a melhor solução para os problemas deste projeto.

2.2 Estado da Arte

2.2.1 Aplicações dos sistemas de monitorização de dados

Os sistemas de aquisição de dados são muito usados hoje em dia para recolher informação sensorial do mundo que nos rodeia. Este tipo de sistemas é usado em diversas áreas.

A temperatura é dos parâmetros mais monitorizados uma vez que tem influência na vida humana. Existem muitos sistemas que fazem a recolha e monitorização deste parâmetro para várias finalidades. Em [16, 17, 6, 18] foram implementados sistemas para monitorizar apenas temperatura em espaços fechados, nomeadamente: num

laboratório [16], numa *greenhouse*¹ [17] e numa arca frigorífica [18]. A monitorização da temperatura pode também ser usada no controlo de sistemas de produção, em [19] são usados sensores de temperatura para monitorizar e controlar a temperatura numa máquina de moldagem de plástico.

Existem, no entanto, sistemas que para além da temperatura fazem recolha de outro tipo de parâmetros, como por exemplo: a humidade, a luz, a pressão, ou a existência de gases no ambiente em estudo. Em [2] para além do sensor de temperatura foi usado um sensor para detetar as condições de luz no laboratório e um sensor para detetar a presença de gases. Com outro objetivo em [5] pretende-se fazer o estudo dos parâmetros da água, este sistema pode ser implementado num rio ou lago para determinar as características da água. Para isso possui um sensor que analisa o espectro da água, um sensor de gás que faz a deteção de CO₂ (Dióxido de Carbono), e um sensor que recolhe dados da temperatura e pressão da água.

2.2.2 Aplicações da *Raspberry Pi*

Com a finalidade de obter uma solução final de baixo custo, as soluções estudadas implementam sistemas que utilizam componentes de baixo custo mas que cumpram com as necessidades de cada solução. Uma vez que foi escolhido a *Raspberry Pi* como *hardware* base para o desenvolvimento deste projeto, as soluções foram também escolhidas tendo em conta o uso deste pequeno computador. A *Raspberry Pi* é um pequeno computador com a capacidade de um computador de secretária, no qual é possível aceder à internet ou ligá-lo a um monitor para ver um filme. Por outro lado tem também a capacidade controlar periféricos ligados nas suas entradas, como por exemplo sensores [1].

São muitas as aplicações em que este computador é usado. No âmbito deste projeto destacam-se as aplicações que tem como objetivo a monitorização da temperatura através de uma página *web* [3, 4, 16, 17, 6, 2]. No entanto existem outras aplicações em que este computador é usado, em [3] é referido a automação em casas uma vez que a *Raspberry Pi* permite o controlo e monitorização de sensores. É possível o controlo de luzes, a monitorização da temperatura, o controlo de portas, a vigilância através de câmaras, etc. [20]

Também em [21] é implementado um projeto na área da automação em casas. A *Raspberry Pi* é usada para vigiar uma porta através de uma câmara que envia

¹*greenhouse* - Estufa com parâmetros controlados para um ótimo crescimento de plantas.

vídeo via internet, depois de detetar a presença de pessoas através de um sensor infravermelho. O utilizador acede à internet e depois de visualizar a entrada da casa através de vídeo consegue abrir a porta através de um botão.

Uma outra aplicação muito procurada para este dispositivo é a implementação de um servidor *web*, possibilitando armazenar dados e criar páginas de internet para a visualização dos mesmos [22, 23, 4, 2].

2.2.3 *Raspberry Pi* como nó sensor

Quando se fala em sistemas de recolha de dados sensoriais para monitorização *web* deve-se falar em WSN (Wireless Sensor Network) ou rede de sensores sem fios. As redes de sensores sem fios são compostas por nós sensor e cada nó sensor é constituído por quatro elementos: unidade sensorial, unidade de armazenamento e processamento, unidade de comunicação de dados, e unidade de alimentação do nó [1]. Na figura 2.1 pode-se verificar como cada unidade se relaciona.

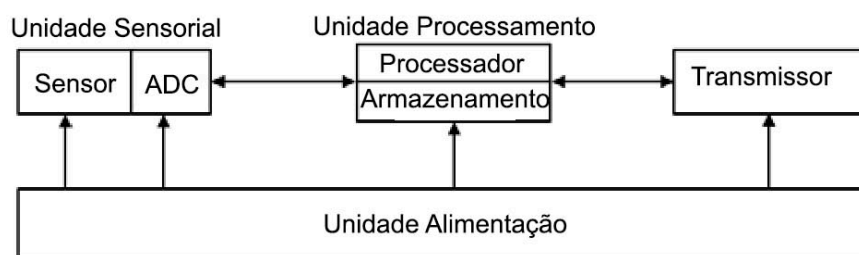


Figura 2.1: Nó sensor [1].

Como já foi referido a *Raspberry Pi* possui a capacidade de controlar periféricos ligados nas suas entradas, como tal é possível implementar nele um nó sensor, uma vez que possui as características necessárias. Em[1] fez-se um estudo da *Raspberry Pi* como nó sensor. Este é comparado com outros dispositivos usados para o mesmo efeito possibilitando verificar as vantagens e desvantagens do seu uso. Um dos objetivos principais de uma rede de sensores é recolher dados do maior número possível de locais, no entanto é necessário ter em atenção o custo final do projeto, uma vez que para a recolha de dados em múltiplos locais é necessário utilizar muitos nós sensor, logo o custo de cada nó não pode ser elevado. O tamanho de um nó sensor é uma característica a ter em atenção, sendo o nó sensor pequeno pode ser colocado em locais diversos com dimensões que apenas permitam o uso de nós pequenos. Na tabela 2.1 é feita uma comparação do tamanho, peso e custo da *Raspberry Pi* com

outros dispositivos semelhantes. Verifica-se que a grande vantagem deste dispositivo é no seu custo, sendo que num projeto em que seja necessário o uso de vários nós sensor, o custo total do projeto vai ser reduzido com a *Raspberry Pi*.

Tabela 2.1: Comparação tamanho, peso, custo[1]

Nome	Tamanho (mm)	Peso (g)	Preço US\$
Raspberry Pi	85.6*53.98*17	45	25-35
MicaZ	58*32*7	18	99
TelosB	65*31*6	23	99
Iris	58*32*7	18	115
Cricket	58*32*7	18	225
Lotus	76*34*7	18	300

Relativamente à capacidade de processamento, a *Raspberry Pi* possui um processador com arquitetura ARM (Acorn RISC Machine) 11, 32 bit e 700 MHz. Das suas características fazem parte também os 256 MB ou 512 MB de memória RAM (Random Access Memory), dependendo do modelo. Uma característica a destacar é o facto de que para funcionar necessita 5V/1A, possibilitando por exemplo que seja alimentado por uma porta USB (Universal Serial Bus). No mesmo artigo[1] fez-se também uma comparação nos dispositivos usados como nó sensor, referente ao processador usado, memória RAM e memória externa. Na tabela 2.2 verifica-se novamente que a *Raspberry Pi* ganha vantagem relativamente aos outros dispositivos, destacando-se na capacidade de memória RAM que possui, uma vez que esta é essencial para o bom desempenho do sistema a implementar.

Tabela 2.2: Comparação processador, RAM, memória externa [1].

Nome	Processador	RAM	Memória externa
Raspberry Pi	ARM BCM2835	256-512 M	2-64 G
MicaZ	ATMEGA128	4 K	128 K
TelosB	TI MSP430	10 K	48 K
Iris	ATMEGA1281	8 K	128 K
Cricket	ATMEL128L	4 K	512 K
Lotus	ARM NXP LPC1758	64 K	512 K

As redes de sensores sem fios são usadas nas mais variadas situações, e como tal, os nós sensor devem ser constituídos por *hardware* flexível e adaptativo que permita a implementação de variados projetos. No entanto, e para diminuir os custos do projeto, os nós sensor devem apenas possuir o *hardware* necessário a cada sistema. Isto faz da *Raspberry Pi* o *hardware* apropriado a qualquer sistema de monitorização e controlo de parâmetros, uma vez que os sensores podem ser conectados nas suas variadas entradas. A *Raspberry* possui entradas e saídas digitais que suportam comunicação série, I²C²(Inter-Integrated Circuit) e SPI³(Serial Peripheral Interface), simplificando a conexão de sensores com características diferentes.

Tal como foi visto anteriormente, um nó sensor necessita de uma unidade de transmissão de dados. A *Raspberry Pi* possui uma porta *ethernet*⁴ que facilita a comunicação com outros dispositivos através da internet, permitindo que numa rede de sensores os nós sensor comuniquem facilmente. Para além da porta *ethernet* existem entradas USB, permitindo a conexão de um adaptador *WiFi* que possibilita aceder sem fios à internet, facilitando a comunicação dos nós sensor nas redes de sensores sem fios.

A *Raspberry Pi* é um computador, e como tal necessita de um sistema operativo. O *Raspbian* é um sistema operativo com base Linux, adaptado a processadores com arquitetura ARM e tem a vantagem de ser grátis, ou seja, não necessita de uma licença paga. O Raspbian é um sistema operativo com um ambiente gráfico muito parecido com *Windows* e com *MAC* (Macintosh Operating System), o que permite uma fácil adaptação do utilizador. Na tabela 2.3 estão referenciados os sistemas operativos usados nos dispositivos comparados anteriormente. Destaca-se o uso do *Raspbian* como sistema operativo usado na *Raspberry Pi*.

Concluindo, a *Raspberry Pi* é um versátil, pequeno e barato computador que tem as características essenciais para ser usado como nó sensor ou como coordenador de uma rede de nós sensor. Tem vantagens em relação aos outros dispositivos analisados pela sua capacidade de processamento, pela sua extensível memória externa que permite armazenar dados e pela sua capacidade de conexão com outros dispositivos. A possibilidade de ser ligado à internet permite que seja usado como servidor *web*, o que lhe dá também a possibilidade de ser usado como nó sensor *web*, ou seja, um nó

²I²C - Protocolo de comunicação série que suporta múltiplos dispositivos e apenas requer dois fios.

³SPI - Protocolo de comunicação série que funciona nos dois sentidos, ou seja, envia e recebe dados ao mesmo tempo.

⁴*ethernet* - Tecnologia que define a arquitetura de uma rede local (LAN (Local Area Network))

Tabela 2.3: Comparação sistemas operativos [1].

Nome	Sistema Operativo
Raspberry Pi	RASPBIAN
MicaZ	TINY OS, MOTE RUNNER
TelosB	TINY OS, SOS, MANTISOS
Iris	TINY OS, MOTE RUNNER
Cricket	TINY OS
Lotus	RTOS, TINY OS

sensor que guarda os parâmetros do sensor e permite que sejam visualizados através da *web*. Em termos de desvantagens, destaca-se o consumo de energia, normalmente varia entre 700 mA e 1A mas depende muito dos periféricos ligados nas suas entradas. A falta de um conversor analógico digital integrado é uma desvantagem, no entanto este problema é facilmente resolvido com componentes externos[1].

2.2.4 Redes de sensores

Depois de analisar os componentes essenciais para a criação de uma rede de sensores, foi feita uma análise de cada solução no que se refere aos componentes utilizados e o modo de funcionamento das redes de sensores.

Em [2] a *Raspberry Pi* recebe informação dos sensores através de uma placa de aquisição de dados que está conectada à *Raspberry Pi* com um cabo RS232, ou seja, os dados são enviados via comunicação série. Os sensores de temperatura (LM35), luz (LDR) e gás são ligados a uma placa que faz a conversão de analógico para digital e comunica os dados para a *Raspberry Pi*. Na placa que adquire os dados dos sensores foi ligada uma ventoinha que atua de acordo com valores estabelecidos para a temperatura. Utilizando a porta *ethernet* os dados são enviados para um servidor *web* possibilitando a sua monitorização usando a internet. É possível a monitorização dos dados em qualquer local, desde que se possua acesso à internet e um dispositivo que o permita. Esta solução está representada por um diagrama de blocos na figura 2.2

De outra forma em [3], não é utilizada nenhuma placa para adquirir os dados

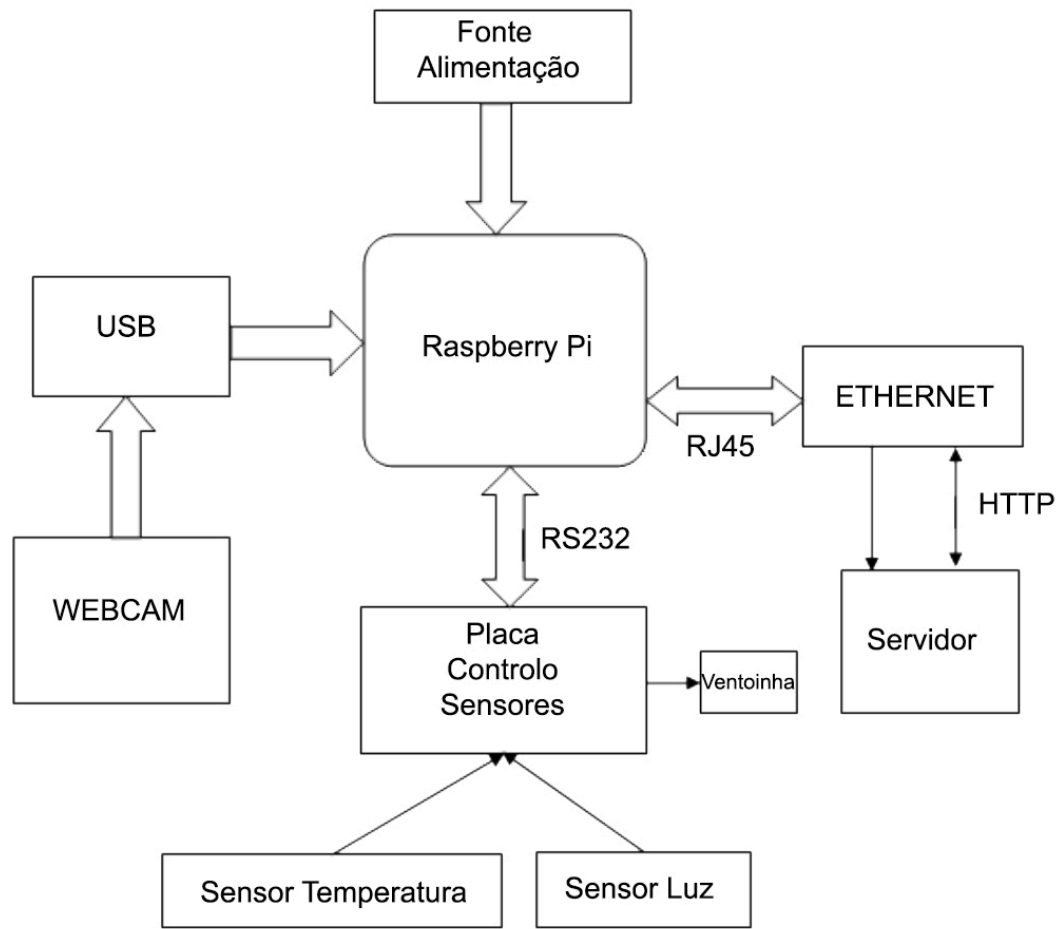


Figura 2.2: Diagrama de blocos [2].

dos sensores. Os sensores são diretamente ligados aos pinos da *Raspberry Pi*. São utilizados um sensor de temperatura (DS8B20), um sensor de humidade (DHT22) e um sensor de pressão (BMP085), que enviam os dados recolhidos por intermédio do protocolo I²C. Na *Raspberry Pi* é implementado um servidor *web* que guarda os dados e através de uma página *web* permite que estes sejam acedidos através de um *browser* num dispositivo com acesso à internet. Na figura 2.3 está ilustrado o diagrama de blocos de todo o sistema.

Existem no entanto redes de sensores que usam outro tipo de comunicação entre o nó sensor e o coordenador da rede. Em [4] foi usado o protocolo de comunicação *Zigbee* na comunicação dos nós sensor com o coordenador da rede.

Neste sistema existem três módulos distintos: um módulo nó sensor, um módulo servidor e um módulo que monitoriza os dados. O nó sensor é constituído por

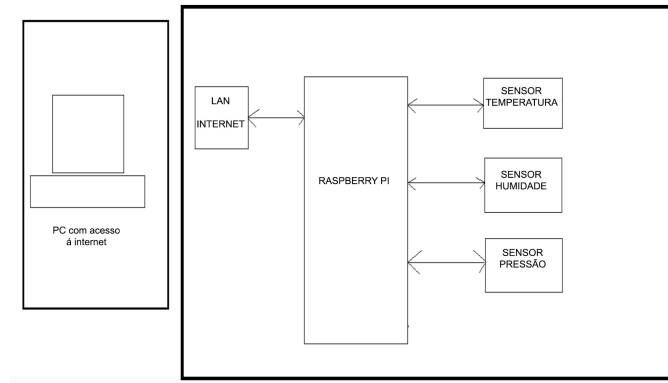


Figura 2.3: Diagrama de blocos [3].

um microcontrolador 89C51, este recebe o sinal digital dos sensores de temperatura (AD590) e gás (MQ2) via I²C convertido por um ADC (Analog-to-Digital Converter) (PCF8591). Ao micro está conectado um módulo *XBee*, este módulo tem implementado o protocolo de comunicação sem fios *Zigbee* especificamente desenvolvido para este tipo de sistemas. É um módulo que tem como principal característica o custo reduzido e o baixo consumo de energia. No nó sensor também é conectado um LCD (Liquid Crystal Display) que permite a visualização dos valores dos sensores. O módulo servidor é constituído por uma *Raspberry Pi* e um módulo *XBee*. A *Raspberry Pi* é configurado como servidor *web* e através do protocolo TCP/IP (Transmission Control Protocol/Internet Protocol) permite que os dados comunicados ao servidor via *XBee* possam ser acedidos pelo módulo que faz a monitorização dos dados. O módulo de monitorização dos dados não é mais do que um PC que acede aos dados via protocolo TCP/IP. Os diferentes constituintes deste sistema estão representados por um diagrama de blocos na seguinte figura 2.4.

Devido ao facto de existirem vários tipos de comunicação na aquisição de dados dos sensores, em [5] foi implementada uma solução para modelar os nós sensor de forma a que o recetor dos dados (computador) não se preocupe com o tipo de comunicação de cada sensor. Existem três sensores implementados e cada um comunica de forma diferente: o espectrómetro (10988MA) envia a sua informação via sinal de vídeo, o sensor de CO₂ (K30) envia os seus dados através de comunicação série UART, o sensor de temperatura e pressão (SCP1000) comunica através de SPI. Para simplificar a programação na *Raspberry Pi*, uma vez que teria de ser implementado um programa que recebesse os diferentes sinais dos sensores, foi usado um microcontrolador (R8C). Cada nó sensor tem um microcontrolador programado

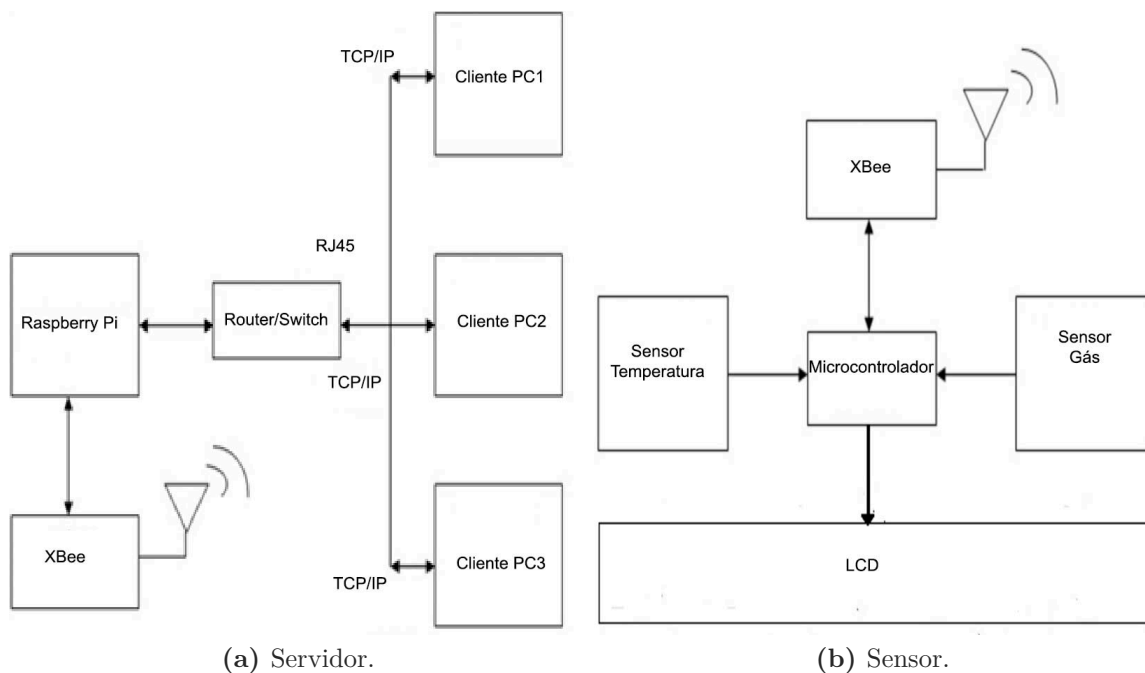


Figura 2.4: Diagrama de blocos [4].

para receber um tipo de sinal de um sensor específico, e comunica via USB com a *Raspberry Pi*. Assim a *Raspberry Pi* tem a tarefa simplificada, uma vez que apenas envia comandos através de uma porta série selecionada a pedir dados de um determinado sensor. O microcontrolador conectado a essa porta recebe o comando e envia os dados do sensor. Na *Raspberry Pi* foi também conectada uma câmara que faz captura de imagens e comunica através do protocolo CSI⁵(Camera Serial Interface). Na figura 2.5 está representado o diagrama de blocos de todo o sistema.

O protocolo *Zigbee* tem sido muito usado nas redes de sensores sem fios, exemplo disso é o modelo implementado em [6]. Neste modelo um módulo *Zigbee* é implementado num microcontrolador ATtiny que comunica com a *Raspberry Pi*. Cada nó sensor é constituído por um sensor de temperatura PT100 e todos os nós sensores estão conectados ao microcontrolador ATtiny. O microcontrolador converte o sinal analógico em digital e envia-o para a *Raspberry Pi* via *Zigbee*. Depois de recebidos, os dados são colocados num servidor *web* para que possam ser monitorizados através de um dispositivo com acesso à internet. Este sistema tem o seu modelo representado pelo diagrama de blocos na figura 2.6.

⁵CSI - Protocolo de comunicação série desenvolvido para utilização com a câmara *Raspberry Pi CSI Camera Interface*.

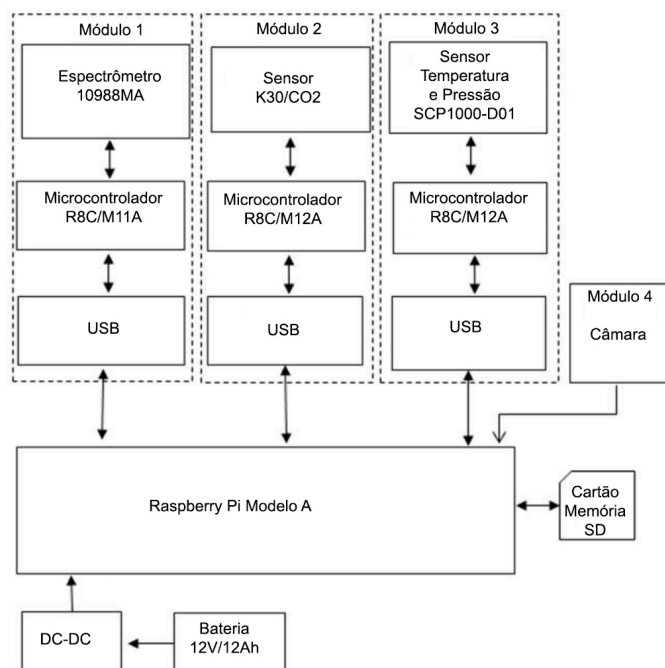


Figura 2.5: Diagrama de blocos [5].

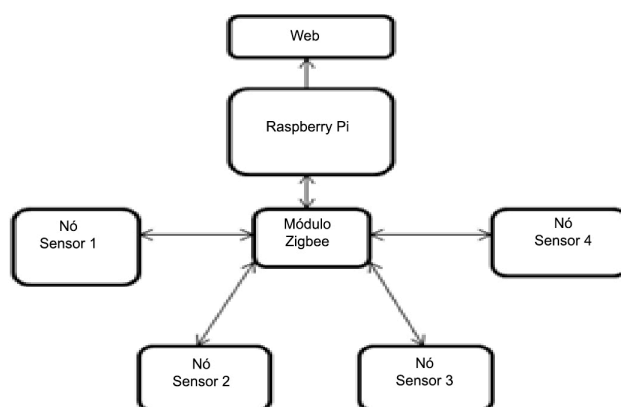


Figura 2.6: Diagrama de blocos [6].

3 Fundamentos Teóricos

Este capítulo tem como objetivo descrever teoricamente os métodos e materiais utilizados nesta dissertação.

3.1 UML

O UML (Unified Modeling Language) é uma linguagem de modelação que tem como principal objetivo padronizar modelos. Nasceu no âmbito da engenharia de *software* com a finalidade de padronizar os sistemas de informação. Ou seja, independentemente das metodologias utilizadas no sistema, o UML utiliza diagramas para que o sistema seja descrito de uma forma geral sem que mencione explicitamente as metodologias utilizadas.

O UML não diferencia linguagens de programação nem processos de desenvolvimento, e independentemente do projeto, das ferramentas utilizadas ou das metodologias apropriadas a cada projeto, mantém a mesma linguagem de modelação.

Do UML, fazem parte um conjunto de diagramas que explicam de uma forma simplificada o funcionamento de um sistema. Os vários tipos de diagramas diferenciam-se em diagramas de estrutura e diagramas de comportamento. Os primeiros dão ênfase ao modo como os vários elementos do sistema se organizam, enquanto que os segundos revelam o modo como os elementos do sistema se comportam.

3.2 Aplicações Nativas e Aplicações *Web*

A evolução da tecnologia nos dispositivos móveis trouxe com ela um novo paradigma referente ao desenvolvimento de aplicações nestes dispositivos. Por um lado temos a abordagem antiga em que se criam aplicações nativas, ou seja, aplicações instaladas nos dispositivos que tiram o máximo partido das suas características. Por outro, devido à facilidade de acesso à *web* e à evolução das linguagens de programação, nasceu uma nova maneira de desenvolver aplicações. Estas aplicações designam-se

por aplicações *web* e distinguem-se por serem apenas constituídas por uma página *web* acedida pelo *browser*¹ instalado no dispositivo.

Por aplicação nativa define-se uma aplicação criada para funcionar num determinado sistema operativo. Tem a vantagem de mais facilmente tirar partido das características do dispositivo, e consequentemente, ser uma aplicação que funciona melhor. Por outro lado, existe a desvantagem de ser necessário criar várias versões da mesma aplicação para os diferentes sistemas operativos.

Uma aplicação *web* caracteriza-se por funcionar através de um *browser* instalado no dispositivo, evitando a necessidade de ser instalada, como é o caso de uma aplicação nativa. O facto de existirem diferentes *browsers* é uma desvantagem, fazendo com que a aplicação tenha de ser desenvolvida de maneira a adaptar-se a todos os *browsers*. No entanto esta desvantagem tem sido atenuada pela evolução das linguagens de programação, mais propriamente do HTML5, que faz com que as aplicações *web* sejam criadas de uma forma padronizada de modo a funcionarem corretamente nos diferentes *browsers*. Relativamente ao aproveitamento das funcionalidades do dispositivo móvel, por exemplo o GPS (Global Positioning System), também as linguagens têm evoluído no sentido de mais facilmente ser possível aceder a essas funcionalidades.

Existem vantagens e desvantagens nos dois tipos de aplicações, no entanto, no âmbito desta dissertação foi escolhido desenvolver uma aplicação *web*. Esta escolha deve-se ao facto de a aplicação ser desenvolvida para funcionar em dispositivos móveis pertencentes à empresa. Dispositivos estes que não são todos iguais e, no caso de serem substituídos, poderão não ser substituídos por dispositivos iguais. Fazendo com que a aplicação *web* possa ser acedida de qualquer dispositivo que venha a ser adquirido.

3.3 Internet

A internet é a maior rede de comunicação mundial, utiliza um conjunto de protocolos de comunicação que ligam redes de computadores em todo o mundo.

Através da Internet são implementados vários tipos de serviços, como por exemplo a *World Wide Web*, mais conhecida por *web*, e o *email*. Estes serviços funcionam de acordo com o modelo cliente-servidor. Os clientes são dispositivos ligados a uma rede e podem ser computadores ou dispositivos móveis. Neste modelo, o cliente

¹*browser* - Programa utilizado para visualizar páginas *web*

faz pedidos de serviços que vão ser fornecidos pelo servidor. O servidor fornece ao cliente funções ou serviços que podem ser por exemplo o acesso à *web*.

A *web* é um conjunto de informação que está disponível na forma de texto ou multimédia. O utilizador pode aceder a esta informação através de um *browser*. Toda a informação disponibilizada está guardada em servidores *web*.

3.4 Linguagens de Programação Web

As páginas *web* que estão acessíveis através de um *browser* são codificadas recorrendo a linguagens de programação. Existem dois tipos de páginas *web*, as estáticas e as dinâmicas. As páginas estáticas apenas dispõem informação que está na própria página. As páginas dinâmicas permitem aceder a informação em tempo real, guardada em bases de dados.

As várias linguagens de programação permitem a criação de páginas *web*. Nesta dissertação foram utilizadas as seguintes linguagens de programação: HTML (HyperText Markup Language), CSS (Cascading Style Sheets), *JavaScript*, PHP (Hypertext Preprocessor) e SQL (Structured Query Language). Foram escolhidas por serem linguagens muito usadas nas páginas *web*. A figura 3.1 ilustra o modo como as várias linguagens de programação se relacionam para criar páginas *web*. Pode-se verificar que enquanto o HTML, o CSS e o *JavaScript* funcionam do lado do *browser*, o PHP funciona do lado do servidor. Neste diagrama destaca-se também o modelo cliente-servidor já referido, o *browser* faz pedidos ao servidor usando HTML, CSS e PHP. Do outro lado, o servidor interpreta o PHP e responde sob a forma HTML e CSS.

3.4.1 HTML5 & CSS3

O HTML é uma linguagem de programação que faz a estruturação dos conteúdos apresentados nas páginas *web*. O aspeto gráfico do conteúdo presente nas páginas *web* é definido pelo CSS e este define-se também com uma linguagem de programação.

O HTML separa e organiza o conteúdo de uma página *web* em vários elementos. Os elementos de uma página podem ser por exemplo o cabeçalho, os parágrafos e as tabelas. A versão mais recente é o HTML5. Com a evolução da tecnologia, os meios de acesso a páginas *web* foram também evoluindo, e hoje em dia é muito fácil ace-

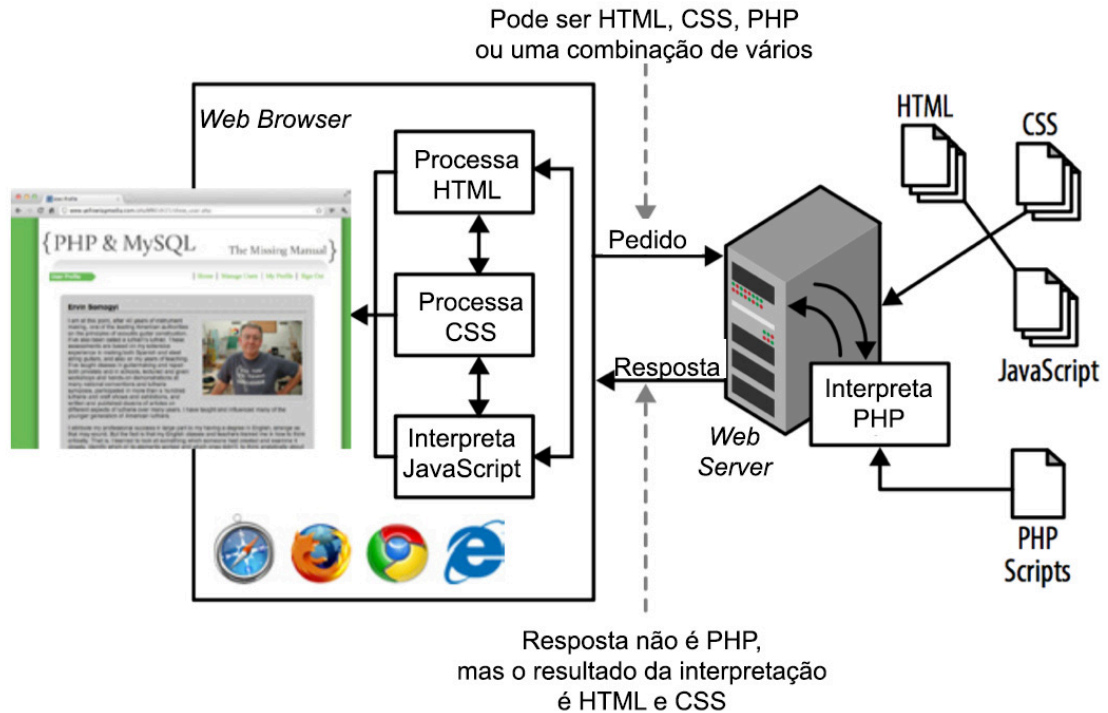


Figura 3.1: Diagrama *web* [7].

der através de dispositivos móveis. Para complementar esta evolução, as linguagens de programação desenvolveram-se, permitindo aproveitar melhor os recursos que os dispositivos nos disponibilizam. A mais recente versão do HTML fornece novas funcionalidades que permitem criar páginas *web* mais facilmente, mas também permite a criação de aplicações *web* com melhores funcionalidades para o utilizador.

O CSS é uma poderosa linguagem que define o aspeto gráfico do conteúdo da *web*. Permite por exemplo definir o tipo de letra, cor da mesma, o fundo da página, etc. O CSS3 é a última versão desta linguagem e possui funcionalidades mais complexas.

3.4.2 JavaScript

A evolução dos *browsers* criou a necessidade de executar *scripts*² do lado do cliente. Para colmatar essa necessidade foi criada a linguagem de programação *JavaScript*. É uma linguagem de programação de alto nível, orientada a objetos e é usada na maioria das páginas *web*. Permite a execução de *scripts* mais rapidamente uma vez

²*scripts* - Linguagem de programação executada dentro de um programa e/ou de outra linguagem, que adiciona funções que são interpretadas em tempo de execução pela aplicação.

que não é necessário passar pelo servidor.

O *JavaScript* permite detetar ações do utilizador, como por exemplo o pressionar de uma tecla, fazendo das páginas *web* mais fáceis de utilizar e de programar. Pelo facto de ser executado no *browser*, o *JavaScript* trouxe também problemas, dos quais se destaca a incompatibilidade nos *browsers*. O IE9 (Internet Explorer 9) veio corrigir a maioria dessas incompatibilidades, no entanto existe uma maneira mais simples de as corrigir. O uso de bibliotecas³ veio simplificar essas tarefas e diminuir esses efeitos.

3.4.2.1 *jQuery*

O *jQuery* é uma biblioteca *JavaScript* muito usada hoje em dia. É uma biblioteca criada com o objetivo de simplificar os *scripts* e resolver problemas existentes na linguagem *JavaScript*, como por exemplo a incompatibilidade em diferentes *browsers*.

O *jQuery* usa CSS para identificar elementos da página *web* e poder manipulá-los através de *scripts*.

3.4.2.2 *jQuery Mobile*

A evolução dos dispositivos móveis que permitem o acesso à *web*, nomeadamente os *tablets* e os *smartphones*, criou a necessidade de desenvolver páginas adaptadas a estes dispositivos. Neste sentido as linguagens de programação também evoluíram, sendo o *jQuery Mobile* uma biblioteca *Javascript* dedicada a este tipo de dispositivos. Quando se usa esta biblioteca pretende-se tirar o máximo partido do dispositivo, fazendo com que, a mesma página ou aplicação *web* funcione da mesma maneira nos diversos dispositivos, sistemas operativos e *browsers* existentes. Ou seja, faz com que a página ou aplicação tenha a maior compatibilidade possível. O *jQuery Mobile* derivou do *jQuery* e usa HTML5 e CSS3, fazendo com que seja uma biblioteca que usa as últimas evoluções das linguagens de programação *web*.

3.4.3 PHP e MySQL

Tal como foi referido anteriormente, existem dois tipos de páginas *web*, as estáticas e as dinâmicas. As páginas *web* dinâmicas diferenciam-se por ser possível aceder a informação que está guardada em bases de dados. Permite por exemplo que na

³bibliotecas - São compostas por código que implementa funções auxiliares de programas ou serviços externos.

mesma página seja possível visualizar diferente conteúdo de acordo com a escolha do utilizador. Os *sites* ou *websites*⁴ dinâmicos são por exemplo os *sites* onde se podem fazer compras *online*. O utilizador cria uma conta e pode fazer compras, encontrando toda a informação sobre os produtos. Todos os dados do utilizador e dos produtos estão armazenados em bases de dados. Um *website* com este tipo de características pode-se designar de aplicação *web* ou *web API* (Application Programming Interface), isto porque o utilizador interage com o *site* através do *browser* permitindo realizar várias tarefas.

Para criar este tipo de *site*, é necessário utilizar PHP e *MySQL*.

O *MySQL* é um sistema que faz a gestão de bases de dados e está instalado no servidor *web*. A base que suporta este programa é o *MySQL server*, que não é mais do que um conjunto de programas que possibilita ao administrador gerir a base de dados. O *MySQL* tal como o nome indica utiliza a linguagem de programação SQL. Esta linguagem de programação é constituída por comandos de gestão de bases de dados que são interpretados pelo *MySQL*. Permite por exemplo, criar uma base de dados nova, inserir dados nessa base de dados ou apagar dados.

O PHP é uma linguagem baseada em *scripts* que faz a ligação entre a página *web* e o *MySQL server*. O PHP funciona do lado do servidor e interage com o *MySQL server*, enviando mensagens codificadas em SQL que vão ser interpretadas e executadas pelo *MySQL*. O código PHP é embebido nas páginas HTML e permite que estas disponibilizem conteúdo dinâmico guardado na base de dados.

3.5 Segurança nas Aplicações Web

As aplicações *web*, devido ao facto de se encontrarem acessíveis via internet, são vulneráveis a ataques que podem pôr em causa os dados nelas armazenados. No âmbito desta dissertação, para proteção dos dados presentes na aplicação desenvolvida, foram utilizados métodos de encriptação para proteger as *passwords* dos utilizadores.

3.5.1 Criptografia

A criptografia faz o estudo de métodos e técnicas pelas quais é possível codificar informação, de maneira a que apenas o recetor da informação, consiga decodificar

⁴*website* - Também designado simplesmente por *site*. É um conjunto de páginas *web* organizadas a partir do mesmo URL (Uniform Resource Locator).

essa informação através de uma chave. Existem vários algoritmos e sistemas criptográficos, dos quais apenas vão ser destacadas as funções de *Hash* criptográfico e os algoritmos simétricos [24, 25].

O método utilizado nesta dissertação tem como princípio o uso de uma chave estática no cálculo da função de *Hash*. Esta função criada vai ser depois usada num algoritmo simétrico. O algoritmo usado foi o *Blowfish* [26].

Este método utiliza várias camadas para codificar as *passwords* dos utilizadores, sendo que para descodificar a *password* é necessário quebrar várias barreiras.

Nesta dissertação não foi aprofundado este tema, uma vez que não era esse o objetivo. Apenas é aqui referenciado o método usado na encriptação das *passwords*.

3.5.2 Vulnerabilidades Web

A OWASP (About The Open Web Application Security Project) é uma organização mundial que se dedica ao desenvolvimento de boas práticas na segurança do *software*. É referida neste trabalho académico, devido ao facto desta organização se preocupar com a segurança das aplicações web, ou seja, elabora estudos acerca das vulnerabilidades a que as páginas de internet estão expostas. Em 2013 apresentou um estudo com as dez vulnerabilidades mais comuns existentes nas páginas de internet [12]. Segundo este estudo, as dez vulnerabilidades mais comuns apresentam-se na tabela 3.1.

Vulnerabilidade
<i>Injection</i>
<i>Broken Authentication and Session Management</i>
<i>Cross-Site Scripting (XSS)</i>
<i>Insecure Direct Object References</i>
<i>Security Misconfiguration</i>
<i>Sensitive Data Exposure</i>
<i>Missing Function Level Access Control</i>
<i>Cross-Site Request Forgery (CSRF)</i>
<i>Using Components with Known Vulnerabilities</i>
<i>Unvalidated Redirects and Forwards</i>

Tabela 3.1: Vulnerabilidades nas aplicações web[12].

3.6 Hardware

3.6.1 Raspberry Pi

A evolução da tecnologia nas últimas décadas permitiu que os computadores fossem cada vez mais baratos e, se tivermos em conta também a miniaturização dos componentes electrónicos, facilmente percebemos o aparecimento de computadores cada vez mais pequenos. Desta gama de computadores surge o *Raspberry Pi* figura 3.2.

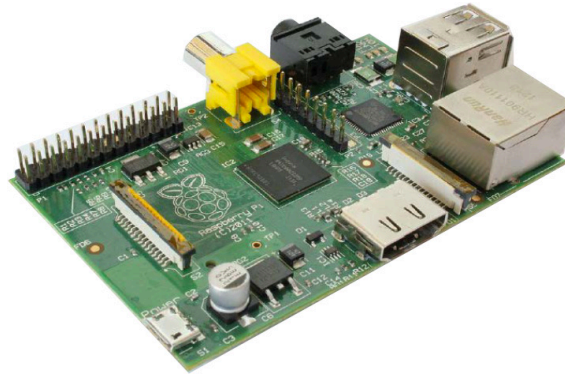


Figura 3.2: *Raspberry Pi* modelo b [1].

O *Raspberry Pi* é um computador de pequenas dimensões e baixo custo com características semelhantes a um computador de secretária. É possível aceder à internet, visualizar vídeos em alta definição e até jogar videojogos. Foi criado como ferramenta de aprendizagem, uma vez que permite o desenvolvimento de múltiplos projetos, nomeadamente na área da eletrónica. Das variadas aplicações e projetos destaca-se a criação de um servidor *web* [4, 23], ou a implementação e monitorização de redes de sensores para ler temperatura e humidade [27, 16][27, 16].

Com ele vem disponibilizado o sistema operativo Linux, e sendo este um sistema operativo de uso livre, isto é, não necessita de uma licença, torna este computador ainda mais popular. Como já foi referido, o processador baseia-se numa arquitetura ARM, e por isso, nem todas as versões dos sistemas operativos Linux podem ser instalados. No caso desta dissertação foi instalado o *Raspbian* [28].

Tendo este computador um sistema operativo incluído, permite que os projetos possam ser desenvolvidos usando múltiplas plataformas e linguagens de programação, podem ser C, C++, *Python* ou Java. No âmbito deste projeto foi escolhida a linguagem de programação *Python*.

Desde o seu lançamento, este computador foi evoluindo e neste momento existem

vários modelos e versões. No âmbito desta dissertação foi usado o Modelo b *Revision* 2.0.

3.6.1.1 Características

A figura 3.3 representa os componentes e o modo como estão organizados na *Raspberry Pi*. O processador incluído na *Raspberry Pi* é um *Broadcom* 700 MHz, funciona a 32 bit, é baseado na arquitetura ARM11 e tem disponível 512 MB de RAM. O sistema operativo é instalado no cartão de memória, que funciona como disco rígido. As entradas USB permitem a conexão de periféricos, como teclado e rato, e/ou de um adaptador *WiFi* de maneira a ser possível conectar à *web*. Por outro lado, tem também uma entrada *ethernet* para poder ligar o dispositivo a uma rede, permitindo também o acesso à *web*. No que toca às saídas de vídeo, existem duas opções, uma saída RCA que permite a ligação a monitores antigos, e uma saída HDMI (High-Definition Multimedia Interface) que permite a ligação a monitores ou televisores mais recentes, para visualização de vídeos em alta definição. No caso de ser usada a saída RCA existe também uma saída de áudio. Os pinos GPIO permitem a ligação de dispositivos electrónicos, como é o caso dos sensores. Estes pinos podem ser usadas também como interface com outros dispositivos, usando os protocolos de comunicação I²C e SPI.

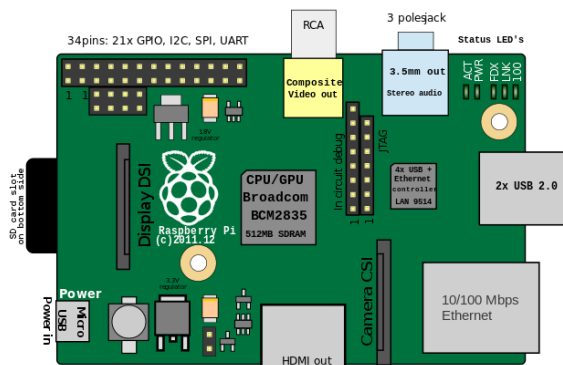


Figura 3.3: *Raspberry Pi* - Componentes [8].

Na figura 3.4 estão representados todos os pinos GPIO e as suas características.

No âmbito deste projeto, optou-se por trabalhar com o *Raspberry Pi* pelo facto de haver um dispositivo disponível para trabalhar, ou seja, não implicou custos. No entanto, como já foi visto anteriormente, o dispositivo cumpre com os requisitos necessários para o desenvolvimento deste projeto. Na tabela 3.2 estão representadas

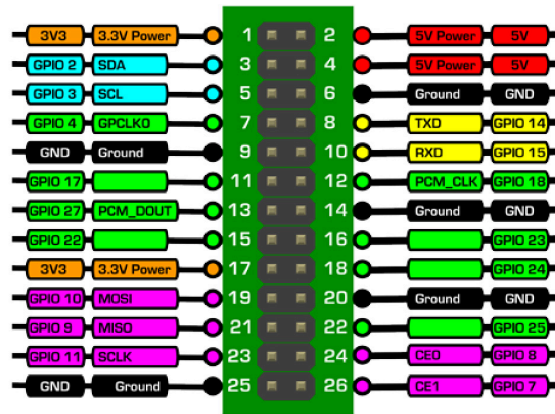


Figura 3.4: Pinos GPIO [1].

as características principais deste dispositivo.

Tabela 3.2: Características *Raspberry Pi*[13].

<i>Raspberry Pi</i> Modelo B - Rev. 2.0	
Chip	Broadcom BCM2835
Processador	ARMv6
Velocidade Processador	700 MHz
Voltagem e Corrente	700mA @ 5V
Tamanho	85x56mm
Memória	512 MB SDRAM @ 400 MHz
Armazenamento	Cartão Memória SD
GPIO	26
USB 2.0	2
Ethernet	10/100 Mbit/s Ethernet - Ficha RJ45
Saída Vídeo	HDMI
Saída Áudio	Analógica - Jack 3.5 mm/ Digital - HDMI

3.6.1.2 *Python*

Python é uma linguagem de programação de alto nível, caracteriza-se por ser uma linguagem multiparadigma, ou seja, suporta programação orientada a objetos mas também programação funcional. Também por isso suporta várias linguagens, como C, C++ ou Java. É muito usada por ter a característica de em poucas linhas de código ser possível expressar conceitos [3].

3.6.2 Módulo *Wifi* para *Raspberry Pi*

O meio de comunicação escolhido para conectar a *Raspberry Pi* à internet foi o *wifi*. Para tal foi necessário utilizar um módulo *wifi* externo, uma vez que a *Raspberry Pi* não possui *wifi* para comunicar.

O módulo utilizado foi o RTL8188CU, representado na figura 3.5. Este módulo é ligado por USB e utiliza o protocolo de rede 802.11b/g/n que permite atingir velocidades de 150 Mbps [9].



Figura 3.5: Módulo wifi RTL8188CU [9].

3.6.3 DHT22

Inicialmente nesta dissertação foi utilizado o sensor de temperatura DTH22. Na figura 3.6 está representado o sensor e suas dimensões em mm. É um sensor de temperatura e humidade e caracteriza-se por um sensor de baixo custo. Possui um *chip* integrado que faz a conversão analógico digital, ou seja, tem a vantagem de não ser necessário um conversor desse tipo. Uma desvantagem deste sensor é ser lento, ou seja, só é possível adquirir valores de temperatura e humidade a cada dois segundos. No entanto nesta dissertação não era necessário ter mais amostragens por segundo, sendo que as restantes características cumpriram os requisitos necessários. Uma vantagem deste sensor é ser pequeno e também ter baixo consumo energético [10]. Na tabela 3.3 estão descritas as especificações do sensor.

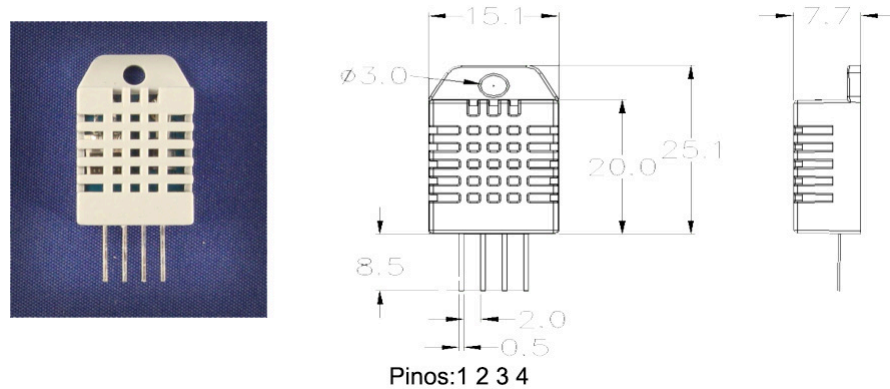


Figura 3.6: Sensor DHT22 e dimensões em milímetros [10].

Tabela 3.3: Características sensor DHT22.

Tensão de alimentação	3,3 - 6 V DC
Corrente	1,3mA-2,1mA
Sinal saída	Digital
Intervalo de medição	Humidade: 0-100% HR/Temperatura:-40°C~125°C
Precisão	Humidade:+-2%HR (Máximo +5%HR)/Temperatura:+-0,2°C
Resolução	Humidade:0,1%HR/Temperatura:0,1°C
Período de leituras	1,7s-2s
Pino 1	VDD
Pino 2	Sinal
Pino 3	Sem função
Pino 4	GND

3.6.4 DS18B20

Posteriormente foi implementado neste projeto o sensor DS18B20 ilustrado na figura 3.7. A escolha deste sensor deveu-se principalmente á facilidade de instalação do mesmo, uma vez que o sensor se apresenta pronto a instalar no local pretendido. O sensor DS18B20 possui *interface* digital *1-Wire*, ou seja, a leitura dos valores de temperatura é feita através de um único fio em formato digital, facilitando a sua implementação.

A resolução da leitura dos valores de temperatura pode ser configurada entre 9 e 12 bits. No entanto o período de leitura varia com a resolução configurada.

Na tabela 3.4 estão designadas as características principais do sensor. Os pinos designados na tabela estão ilustrados na Figura 3.7b [14].

Comparativamente com o sensor anteriormente usado (DHT22), o sensor DS18B20 consegue ter maior resolução na leitura, tem um período de leitura mais rápido e

tem uma maior facilidade na sua instalação, uma vez que possui um cabo pronto a instalar.

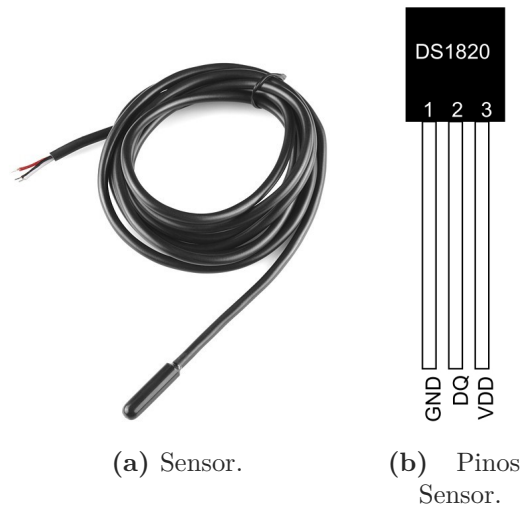


Figura 3.7: Sensor DS18B20

3.6.5 *Raspberry Pi* LCD 1602 Add-on

Para poder visualizar informação do programa a ser executado na *Raspberry Pi*, foi implementado um módulo LCD. Este módulo tem como base um LCD e utiliza os pinos GPIO da *Raspberry Pi* para controlar o LCD. Para além do LCD, possui também cinco botões que podem ser usados, por exemplo, para controlar o cursor no visor ou para aceder a opções que possam ser implementadas. Este módulo possui também extensões dos pinos SDA, SCL, TX, RX e GPIO 7 da *Raspberry Pi* [11]. Assim sendo, é possível usar estes pinos como interface com sensores. No âmbito desta dissertação, o pino GPIO 7 foi usado como entrada para os sensores de temperatura. Na figura 3.8 é ilustrado o módulo referido anteriormente.

Tabela 3.4: Características sensor DS18B20 [14].

(a) Características.

Interface	1-Wire
Intervalo de medição	-55 °C a 125 °C
Precisão	+/- 0.5 °C no intervalo -10 °C a 85 °C
Tensão de alimentação	3.0 V a 5,5 V
Pino 1	GND
Pino 2	Sinal
Pino3	VDD

(b) Resolução.

Bits	Resolução	Período de leitura
9	0.5 °C	93.75 ms
10	0.25 °C	187.5 ms
11	0.125 °C	375 ms
12	0.0625 °C	750 ms

**Figura 3.8:** *Raspberry Pi* LCD 1602 Add-on[11].

4 Metodologias

Este capítulo tem como objetivo descrever todos os passos efetuados e métodos utilizados no desenvolvimento desta dissertação. Vão ser analisadas as partes correspondentes ao *software* e ao *hardware*.

4.1 Visão Geral do Sistema

Na figura 4.1 está representado o diagrama geral deste projeto. Pode-se verificar como todos os componentes comunicam entre si para que o sistema funcione. De realçar o facto de os dois módulos comunicarem entre si via *wifi*.

A *Raspberry Pi*, componente de controlo do sistema, recebe informação dos valores de temperatura de dois sensores. Nela está conectado um LCD onde são visualizadas informações como a data, a hora, e os valores de temperatura dos dois sensores. O módulo *wifi*, conectado também à *Raspberry Pi*, é um dispositivo que lhe permite comunicar via *wifi*. É através desta comunicação que os dados recolhidos dos sensores são enviados para o servidor *web*.

A aplicação *web* acede aos dados armazenados no servidor *web* e permite que estes dados sejam visualizados através de dispositivos com a capacidade de se ligarem à internet, como é o caso dos computadores e *smartphones*. Para além dos dados dos sensores, a aplicação *web* possui um conjunto de utilidades que permite ao utilizador consultar dados provenientes do sistema de faturação da empresa, como é o caso dos artigos e informações sobre os clientes.

4.2 Hardware

Relativamente ao *hardware* utilizado neste projeto, é de realçar o modo como os vários componentes foram ligados entre si. Foram utilizados os seguintes componentes:

- *Raspberry Pi* modelo B, rev. 2.0.

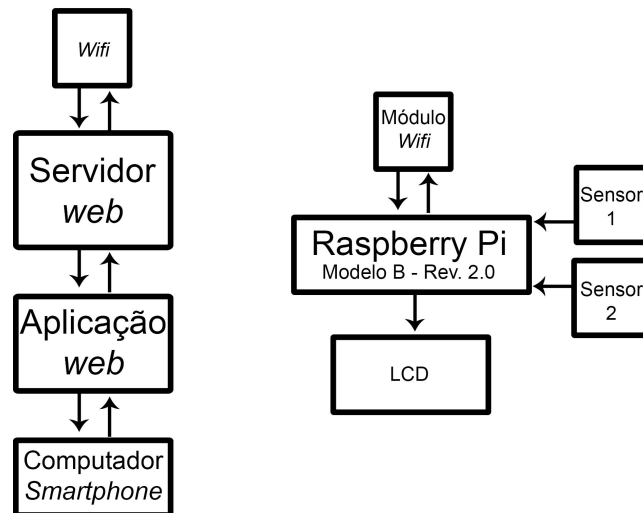


Figura 4.1: Diagrama geral do sistema.

- Módulo LCD1602 para *Raspberry Pi*.
- Módulo *Wifi* RTL8188CU.
- Dois sensores DS18B20.
- Resistência 4.7 K Ω .

Na figura 4.2 verifica-se que o módulo LCD1602 é acoplado perfeitamente na *Raspberry Pi*, sendo que, possui ainda ligação com o pino GPIO número sete da *Raspberry Pi*, onde foram conectados os sensores de temperatura.



Figura 4.2: Módulo LCD1602 acoplado na *Raspberry Pi*.

A figura 4.3 representa o esquemático da ligação entre a *Raspberry Pi* e os sensores. Na *Raspberry Pi* está representada a entrada do pino GP7, onde são ligados em paralelo os dois sensores de temperatura. Destaca-se ainda o uso de uma resistência

de *pull-up* que auxilia a leitura dos valores de temperatura. É de realçar que a ligação dos dois sensores no mesmo pino é possível devido ao uso de uma biblioteca que faz a gestão dos sensores ligados no mesmo pino. Cada sensor possui uma referência única e é possível ler a temperatura de determinado sensor com o auxílio da biblioteca. A biblioteca em questão designa-se de *W1thermsensor* [29].

Para finalizar, é de referir que o módulo *Wifi* RTL8188CU é ligado através das portas USB presentes na *Raspberry Pi*.

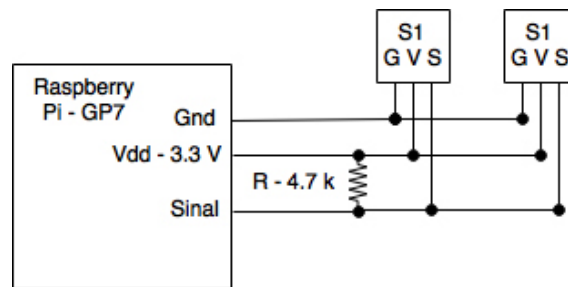


Figura 4.3: Esquemático elétrico de ligação do *hardware*.

4.3 Software

4.3.1 Raspberry Pi

Tal como já foi referido no capítulo 3, o sistema operativo a funcionar na *Raspberry Pi* é o *Raspbian*. Tal como outros sistemas operativos, este permite o desenvolvimento de programas em várias linguagens de programação. O programa desenvolvido para ser executado na *Raspberry Pi* foi elaborado na linguagem *Python*.

O programa desenvolvido tem como tarefas principais as seguintes: ler e escrever no LCD a data e hora, ler os valores de temperatura dos sensores, enviar valores de temperatura para o servidor *web* e enviar alertas via *email* sobre alterações nos valores de temperatura. Este programa não é mais do que um *script* escrito na linguagem *Python* que está continuamente a ser executado. Para evitar que o programa termine a sua execução no caso de uma falha de energia, foi programado para que comece a sua execução quando a *Raspberry Pi* é iniciada. A *Raspberry Pi* possui uma ferramenta que permite agendar o início de programas, possibilitando deste modo, por exemplo, que um programa seja executado em determinado dia da semana. Neste caso, esta ferramenta foi usada para agendar a execução do programa sempre que a *Raspberry Pi* reinicie. Esta ferramenta tem o nome de CRON

e é executada através da linha de comandos.

A figura 4.4 representa o fluxograma do programa que foi desenvolvido. Pode-se verificar que existem três tarefas principais a ser executadas já referidas anteriormente. O programa pode ser interrompido via linha de comandos através do teclado.

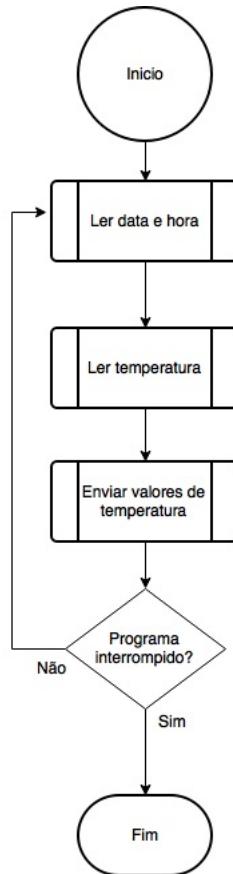


Figura 4.4: Fluxograma do programa executado na *Raspberry Pi*.

Cada uma das principais tarefas estão ilustradas por fluxogramas na figura 4.5. Na figura 4.5a está ilustrada a tarefa que tem como finalidade obter os valores de data e hora e escrever os mesmos no LCD. De seguida, na figura 4.5c é feita a leitura dos valores de temperatura dos sensores. Destaca-se neste fluxograma a existência de um processo que faz a verificação dos valores de temperatura, caso esses valores não estejam de acordo com os valores pré-definidos, um *email* de alerta é enviado para o utilizador, para este verificar a causa deste desvio nos valores e corrigir o problema. No entanto, existe um passo intermédio de verificação da ligação à internet antes do envio do *email*, para que não surjam problemas de ligação. O processo de envio do *email* de alerta está ilustrado na figura 4.5d por um fluxograma. Por fim, os

valores da temperatura são enviados juntamente com os valores da data e hora para o servidor *web* via URL. Uma página *web* do tipo PHP, recebe os valores e armazena-os no servidor para depois serem consultados na aplicação *web*. Esta tarefa é ilustrada na figura 4.5b. De referir também que nesta tarefa também é incluído um passo para verificação da ligação à internet, sendo que os valores apenas são enviados caso a ligação funcione sem problemas.

4.3.2 Aplicação *web*

4.3.2.1 Páginas *web*

A aplicação *web* desenvolvida, não é mais do que um conjunto de páginas de internet que interagem entre si, fornecendo ao utilizador um conjunto de ferramentas.

No total foram desenvolvidas quinze páginas de internet acedidas de acordo com as escolhas do utilizador. Cada página de internet tem funções e características distintas criadas de acordo com os objetivos desta dissertação. Seguidamente vai ser feita uma análise dos requisitos que cada uma das páginas de internet deve ter. As páginas criadas são as seguintes:

- *Login*.
- Inicial.
- Contas Corrente.
- Todas as Contas Corrente.
- Conta Corrente do Cliente.
- Clientes.
- Informação do Cliente.
- Fatura.
- Artigos.
- Famílias.
- Utilizador.

- Atualizar.
- Registrar.
- Arcas.
- *Logout*.

A página “*Login*” foi criada com o propósito de o utilizador inserir os dados de identificação para poder ter acesso às restantes páginas. Esta página apenas permite o acesso de utilizadores previamente registados no sistema. Existem dois tipos de utilizador, o administrador que tem acesso a todas as páginas, e o utilizador comum, que apenas tem acesso a determinadas páginas.

Depois do acesso autorizado na página “*Login*” é apresentada a página “Inicial”. A partir desta página consegue-se aceder à maioria das restantes páginas com as diversas funções. Nesta página acede-se às seguintes páginas: “Contas Corrente”, “Artigos”, “Clientes”, “Atualizar”, “Registrar”, “Arcas” e “*Logout*”. Destaca-se que apenas o administrador tem acesso às páginas “Atualizar”, “Registrar” e “Arcas”.

Uma vez escolhida a página “Contas Corrente”, é apresentada uma listagem dos clientes, com respetivo número e nome associado. É possível pesquisar por nome ou número de cliente e encontrar o cliente pretendido. De realçar o facto que cada utilizador tem um número atribuído. Esse mesmo número é também atribuído a cada cliente, permitindo que a listagem de clientes visualizada, seja apenas dos clientes atribuídos a cada utilizador. O administrador tem acesso a todos os clientes do sistema. Nesta mesma página, o utilizador escolhe um cliente e é direcionado para a página “Contas Corrente do Cliente”. Porém, o administrador do sistema tem acesso também à página “Todas as Contas Correntes”.

A página “Todas as Contas Correntes” permite visualizar uma lista com todos os documentos de todos os clientes. Cada documento é referenciado com uma designação (fatura ou nota de crédito), data de emissão e data de vencimento do documento, código do cliente, valor do documento e saldo total do cliente em questão. Esta página foi criada com o objetivo de filtrar valores por documento, uma vez que, muitas vezes existe o problema de serem feitas transferências para a empresa sem qualquer designação. Fazendo uma pesquisa por valor, consegue-se descobrir a que documento e cliente pertence o valor transferido.

A página “Conta Corrente do Cliente” contém informação detalhada dos documentos emitidos ao cliente escolhido tal como na página descrita anteriormente. No conteúdo desta página estão presentes duas funções para o utilizador:

- Gravar a conta corrente num documento PDF (Portable Document Format).
- Guardar e enviar por *email* o documento PDF.

Na página “Clientes” visualiza-se uma listagem de todos os clientes, esta permite a pesquisa por nome ou número de cliente. O utilizador escolhe o cliente pretendido e é direcionado para a página “Informação do Clientes”.

A página “Informação do Cliente” tem o objetivo de permitir visualizar informação detalhada de cada cliente. Nela, está disponível para consulta a morada do cliente, conta corrente e faturas emitidas ao cliente. Nesta página, o utilizador escolhe uma fatura do cliente e através dessa escolha é direcionado para a página “Fatura”.

Na página “Fatura”, o utilizador consulta a informação detalhada da fatura escolhida. Essa informação é composta pelos artigos mencionados na fatura, quantidade, preço e desconto dos mesmos. A informação do valor da fatura é também visualizada.

Quando o utilizador escolhe a página “Artigos”, uma listagem de todos os artigos existentes é visualizada. Nesta listagem cada artigo é referenciado por um código, nome, IVA correspondente, preço de venda, quantidade disponível e unidade de faturação. Cada artigo pode ser pesquisado por nome ou código. Nesta página o utilizador consegue aceder à página “Famílias”.

Quando acedida, a página “Famílias” dispõem a informação presente na página “Artigos”, no entanto, essa informação é dividida por famílias. A cada artigo é sempre atribuído um código e nome de família, permitindo agrupar os diversos artigos, facilitando a pesquisa.

Através da página “Utilizador” é possível alterar a *password* do utilizador.

A página “Atualizar”, apenas acedida pelo administrador, possui a função de atualizar a base de dados. A base de dados constituída pelos clientes, pendentes, artigos e faturas é atualizada através de ficheiros do tipo CSV (Comma-Separated Values), gerados no programa de faturação da empresa. Estes ficheiros são gravados no servidor *web*, permitindo que a aplicação *web* interage com eles para obter a informação a colocar nas páginas de internet. Esta página, tem também a opção de enviar para todos os utilizadores do sistema um ficheiro em formato PDF. Este ficheiro contém informação dos dias de atraso de pagamento que os clientes têm em determinado documento. O objetivo desta opção é alertar os utilizadores para clientes com muitos dias de atraso nos pagamentos. O ficheiro apenas faz uma listagem dos clientes associados a cada utilizador com mais de noventa dias de atraso nos pagamentos, sendo que, os clientes com mais de cento e vinte dias de atraso nos pagamentos são listados a vermelho.

A página “Registar” foi criada com o objetivo de o administrador registar novos utilizadores no sistema.

Na página “Arcas” é possível visualizar dois gráficos de temperatura das duas arcas instaladas. Possibilita ao utilizador escolher um dia específico para visualizar a temperatura nesse dia.

Por fim, a página “Logout” permite ao utilizador fechar a sessão na aplicação *web*.

4.3.2.2 Diagrama *use case*

Os diagramas *use case* são importantes para definir o comportamento de um sistema. A sua função é definir a comunicação entre os atores e o sistema. No âmbito desta dissertação, foram classificados como atores da aplicação *web*, o administrador e o utilizador comum do sistema. Na figura 4.6 é representado o diagrama *use case* da aplicação *web*.

De uma forma muito intuitiva, o interface entre os utilizadores do sistema é expresso no diagrama, neste é possível identificar as várias funcionalidades que a aplicação *web* possui. Os *use cases* ilustram as página *web* e as funções presentes em cada uma delas.

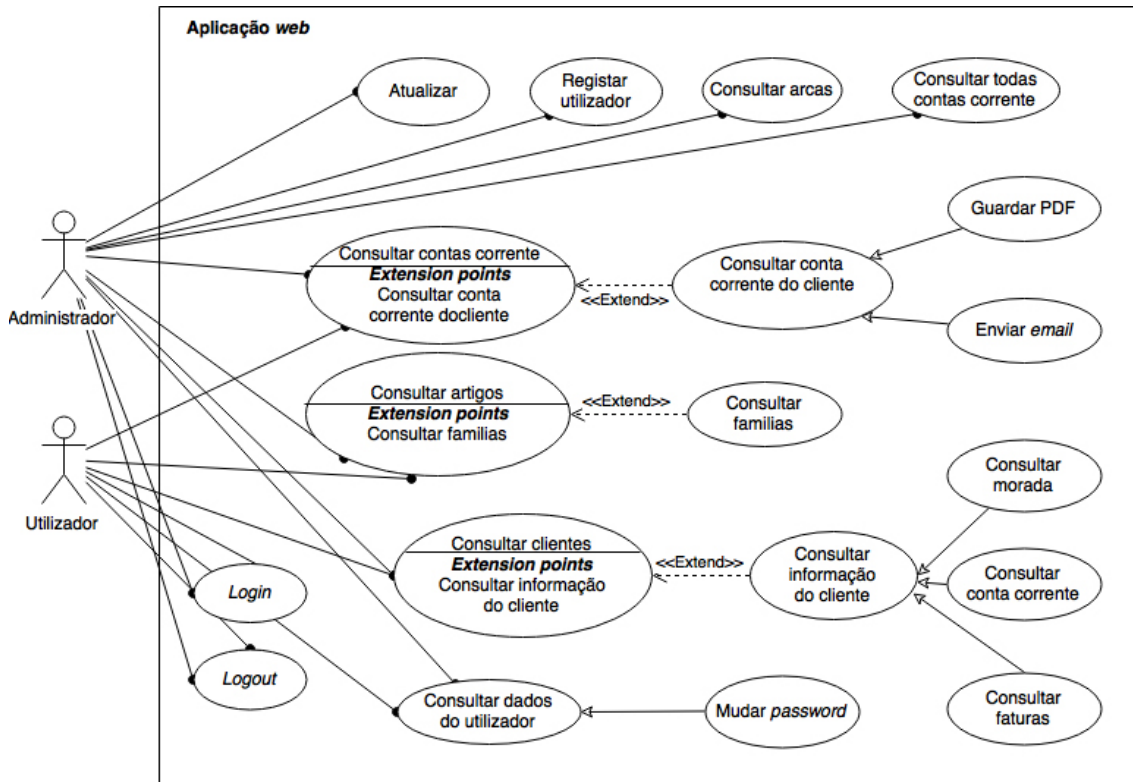


Figura 4.6: Diagrama *Use Case*

4.3.2.3 Diagrama de estados da aplicação web

O diagrama de estados representado na figura 4.7 ilustra todos os estados da aplicação, sendo que cada estado representa uma página *web* desenvolvida. São ilustradas também as transições existentes entre as páginas. No diagrama estão as quinze páginas de internet representadas pelos estados. Consegue-se constatar através de uma análise ao diagrama, a existência de cinco estados principais, sendo que todos eles partilham um interface comum. Isto é, quando o utilizador escolhe um destes estados, tem a possibilidade de nesse mesmo estado, aceder aos outros estados principais.

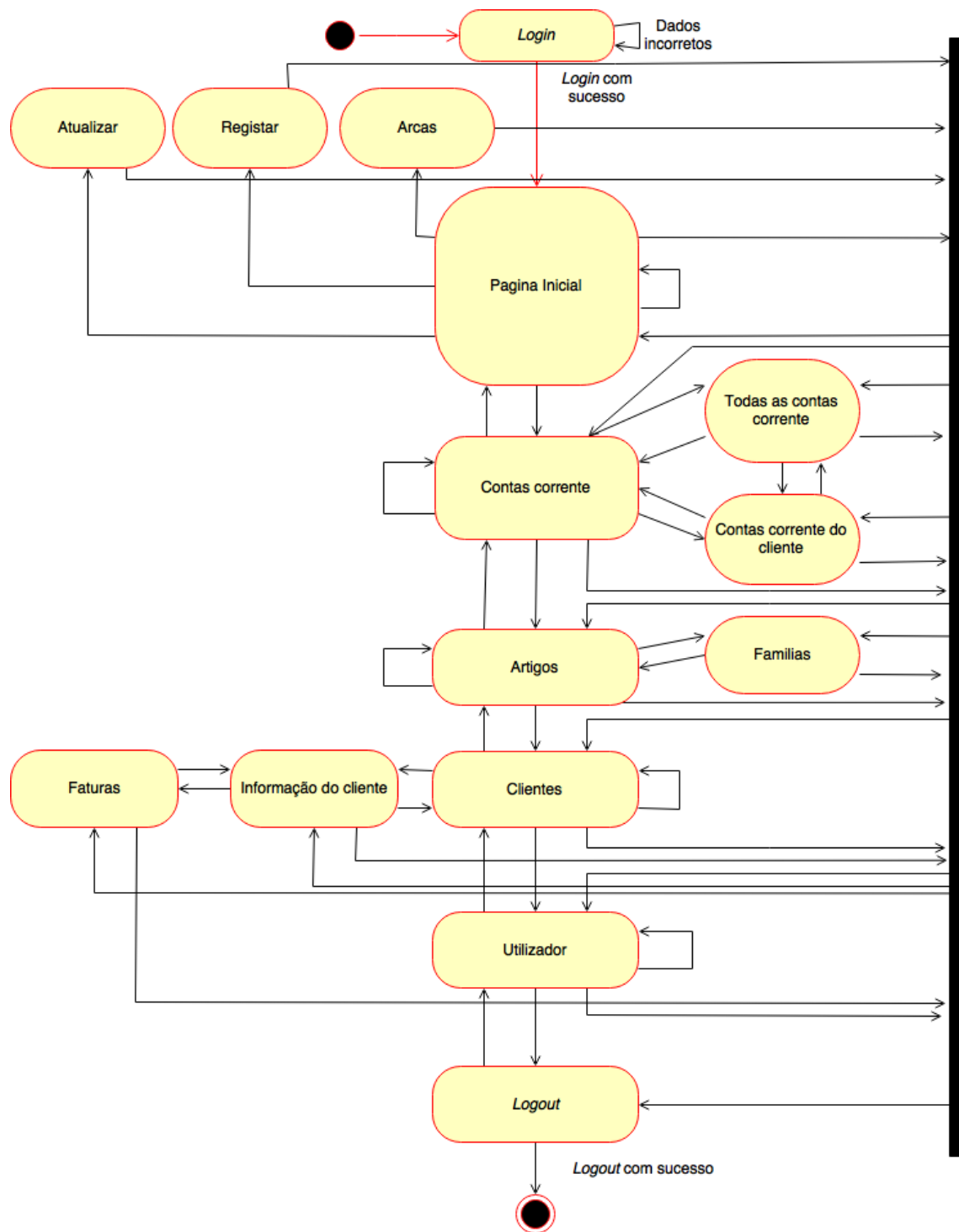


Figura 4.7: Diagrama de estados da aplicação web

4.3.2.4 Arquitetura

A estrutura da aplicação tem como base HTML5 e CSS3, e utiliza a biblioteca *jQuery Mobile*, com o objetivo de criar uma aplicação que seja compatível com a maioria dos dispositivos e que possa ser acedida pela maioria dos *browsers*. Assim sendo, características distintas como o tamanho, sistema operativo e *browser* dos dispositivos utilizados, não influenciam o desempenho da aplicação. Com isto pretendeu-se que a aplicação possa ser acedida de qualquer dispositivo. Por outro lado, a aplicação faz uso da linguagem PHP que funciona do lado do servidor *web*. Esta linguagem, permite à aplicação aceder á base de dados onde está armazenada toda a informação necessária. A base de dados é gerida por um programa designado de *MySQL*, este está instalado no servidor *web* e permite de uma forma simples gerir e armazenar toda a informação.

Na figura 4.8 está representado um diagrama da aplicação *web* desenvolvida. Do lado do *browser* são processadas as linguagens HTML e CSS, estas linguagens são responsáveis pela estruturação e grafismo das páginas de internet, destacando-se o facto de a aplicação ser redimensionada automaticamente de acordo com o tamanho do dispositivo utilizado. O *jQuery Mobile* é responsável por adicionar funcionalidades à aplicação *web*, permitindo uma maior facilidade no interface quando a aplicação é acedida através de *Smartphone* ou *Tablet*.

Quando é necessário visualizar informação guardada na base de dados através da página de internet, o *browser* faz pedidos ao servidor *web*. Este, através da linguagem PHP acede ao *MySQL*, responsável por gerir a base de dados. De referir que, nestes casos as páginas de internet são designadas páginas dinâmicas.

Todos os ficheiros gerados, responsáveis pela execução da aplicação *web* estão armazenados no servidor *web*.

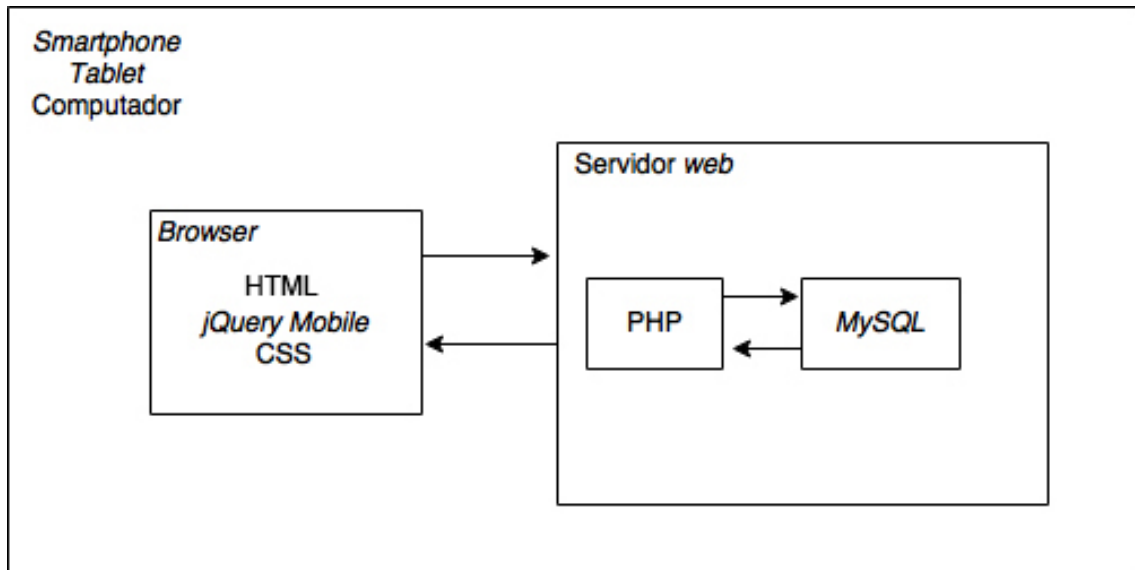


Figura 4.8: Diagrama da aplicação *web*.

4.3.2.5 Base de dados

A figura 4.9 representa o diagrama EER (Extended ER) da base de dados desenvolvida. O diagrama EER é uma versão avançada do diagrama ER (Entidade Relação) criado para ser compatível com as bases de dados mais avançadas. Os diagramas ER são modelos que descrevem o relacionamento entre as entidades de uma base de dados.

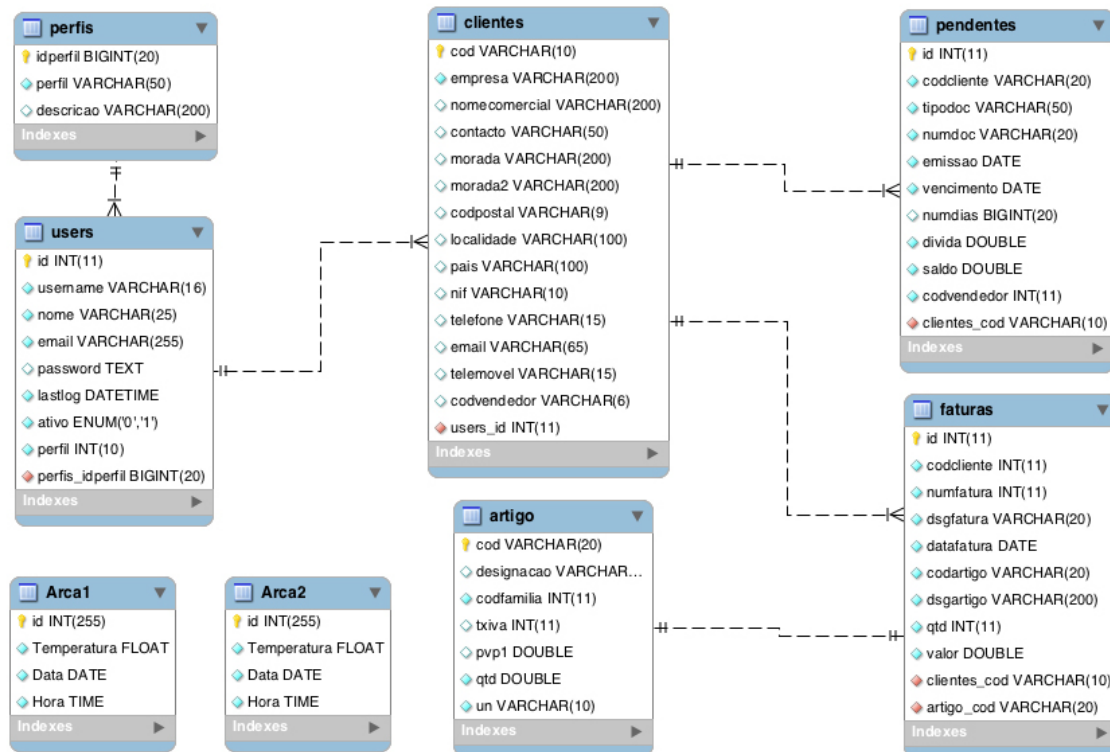


Figura 4.9: Diagrama EER.

Os dados presentes na base de dados são inseridos pelo utilizador da aplicação *web*, mais propriamente pelo administrador. Porém, os dados presentes nas tabelas referentes às arcas frigoríficas, são inseridos automaticamente pelo programa que funciona na *Raspberry Pi*. De realçar que os dados das temperaturas não tem qualquer relação com as restantes tabelas.

A inserção dos dados nas restantes tabelas é feita através de ficheiros do tipo CSV guardados previamente pelo utilizador. Estes ficheiros são provenientes do sistema de faturação da empresa, onde já existia uma base de dados. As tabelas atualizadas com os dados provenientes do sistema de faturação são:

- Clientes: representa todos os clientes.
- Artigo: representa todos os artigos.
- Fatura: representa as faturas dos clientes.
- Pendentes: representa os valores para a conta corrente do cliente.

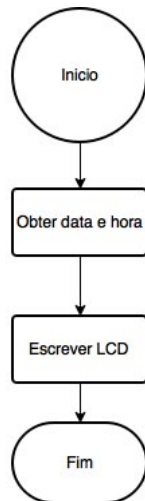
A tabela “*Users*” é atualizada pelo administrador da aplicação *web*.

4.3.2.6 Segurança

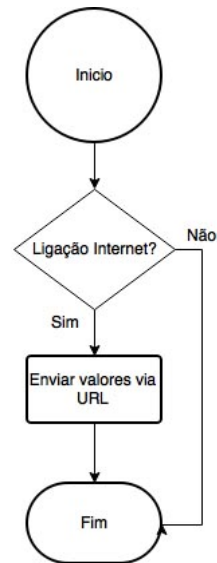
A segurança das páginas de internet é um aspeto importante que deve ser considerado. Uma boa prática de programação é a encriptação das *passwords* dos utilizadores das aplicações *web*. Concretamente, na aplicação desenvolvida no âmbito desta dissertação, foi utilizado o seguinte método para encriptar a *password* de cada utilizador:

- Primeiro foi criada uma chave única que servirá para posterior codificação da *password*, sendo que esta chave é gerada aleatoriamente e não é mais do que um conjunto de caracteres. Como medida de segurança essa chave foi depois guardada num ficheiro no servidor *web*. No entanto, para proteger esta informação, o acesso a este ficheiro foi bloqueado através do *htaccess*. O *htaccess* é um ficheiro de configuração dos diretórios do servidor *web*. Fazendo com que todos os ficheiros a ele associados não possam ser acedidos, estando ocultos no servidor *web*. É de salientar que a chave criada é utilizada em todas as *passwords* inseridas no sistema.
- Segundo, é utilizada uma função de *hash* para encriptação da *password*. A função devolve um *hash* construído pela *password* inserida pelo utilizador e pela chave de segurança criada anteriormente.
- Terceiro, é criado o *salt* para posterior aplicação do algoritmo *blowfish*. O *salt* não é mais do que um conjunto de caracteres criado aleatoriamente, no entanto, para cada *password* é criado um *salt* diferente.
- Para finalizar, é utilizado o algoritmo *blowfish* para criação do código que vai ser armazenado na base de dados. O *blowfish* é um algoritmo simétrico e utiliza o *hash* e o *salt* criado anteriormente.

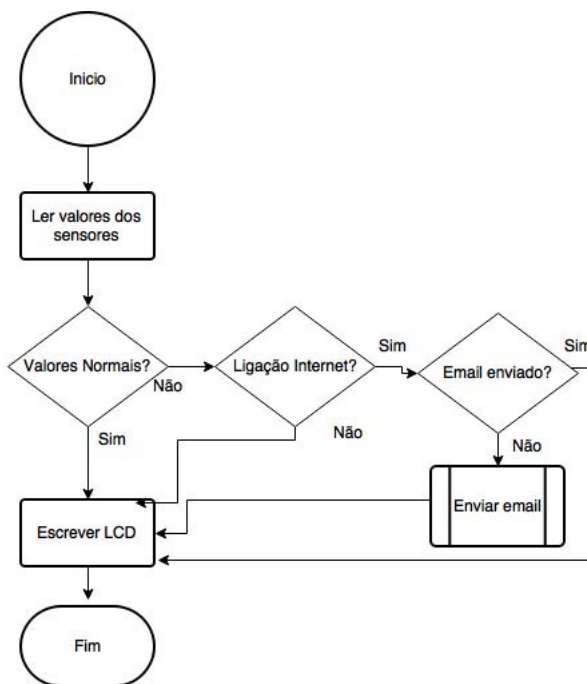
No entanto, existem vulnerabilidades na internet, que vão para além da segurança da *password* que permite o acesso às aplicações *web*. Para verificar, se a aplicação desenvolvida cumpre com os requisitos de segurança, foi efetuada uma análise da aplicação em relação às vulnerabilidades existentes na *web*. Esta análise foi feita com recurso a um *software* de análise de vulnerabilidades *web*. Este *software* de nome *detectify*, faz uma análise às principais ameaças a que as páginas de internet são expostas, tendo como base um estudo realizado pela OWASP em 2013 que enumera as dez principais vulnerabilidades a que as páginas de internet estão expostas.



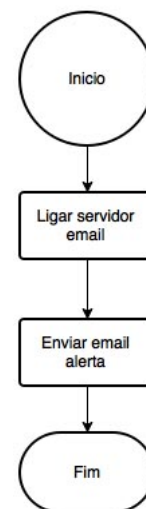
(a) Fluxograma para leitura da data e hora.



(b) Fluxograma para envio dos valores de temperatura via URL



(c) Fluxograma para leitura da temperatura dos sensores.



(d) Fluxograma para envio de email de alerta.

Figura 4.5: Fluxograma das tarefas principais.

5 Resultados e Discussão

No presente capítulo vão ser apresentados os resultados do trabalho efetuado, sendo que, sempre que seja oportuno, esses resultados vão ser discutidos de acordo com os objetivos traçados.

5.1 *Hardware*

O *hardware* desenvolvido no presente trabalho académico, teve como principal função recolher os dados dos sensores colocados nas arcas frigoríficas e enviar os mesmos, via *wifi*, para a base de dados alojada no servidor *web*. Por *hardware* designa-se a *Raspberry Pi*, o módulo LCD, o módulo *Wifi* e os sensores de temperatura. A montagem dos mesmos está ilustrada na figura 5.1, onde se verifica que foi necessário a implementação em *Proto Board* do circuito de ligação dos sensores. Este circuito é apenas composto por uma resistência e por fichas onde estão conectados os sensores. Sendo que, os dois sensores são depois ligados ao módulo LCD, que possui interface com o pino GPIO da *Raspberry Pi*. O acondicionamento da *Raspberry Pi* foi efetuado com recurso a uma caixa fabricada especificamente para este modelo.

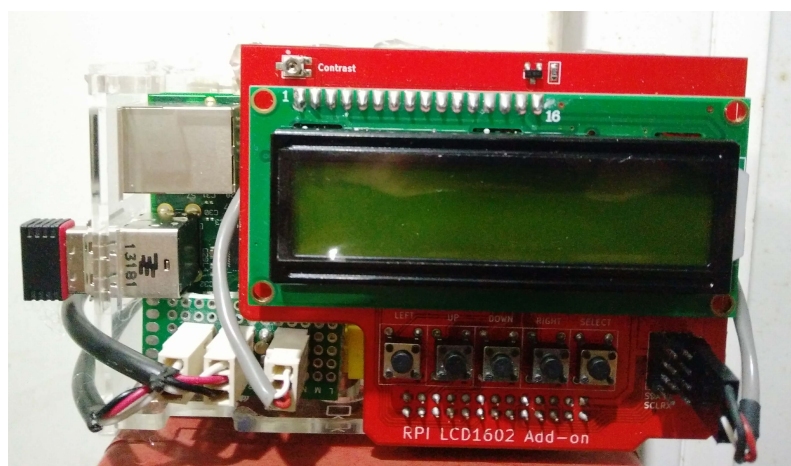


Figura 5.1: Montagem final do *hardware*.

5.2 Aplicação *Web*

5.2.1 Compatibilidade da aplicação *web*

Dos objetivos enumerados no primeiro capítulo, a compatibilidade da aplicação *web* com diversos *browsers*, sistemas operativos e dispositivos com tamanhos de ecrã diferentes vai ser analisada nesta secção.

No âmbito desta dissertação foram considerados para a análise de compatibilidade os *browsers* mais usados hoje em dia:

- *Microsoft Edge*
- *Safari*
- *Google Chrome*
- *Firefox*
- *Opera*

Em todos eles todas as funcionalidades implementadas funcionam como o previsto nos objetivos planeados inicialmente.

Relativamente aos sistemas operativos em que a aplicação foi acedida, destacam-se os seguintes:

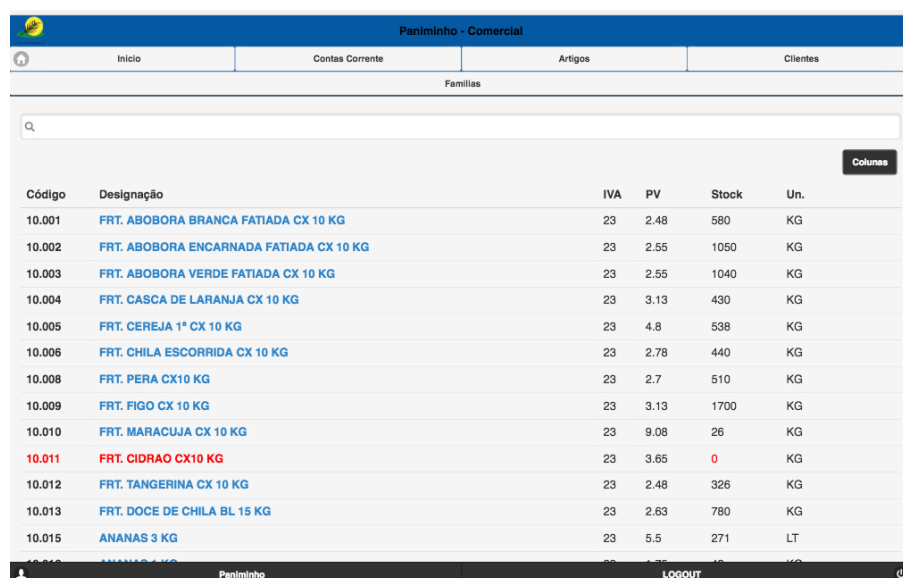
- *Microsoft Windows*
- *IOS*
- *MacOS*
- *Android*

Neste aspeto, a compatibilidade entre aplicação *web* e sistema operativo é também garantida. Em todos os sistemas analisados, acedendo através de diferentes *browsers* à aplicação, não foram detetados problemas de compatibilidade, todas as funcionalidades funcionam corretamente.

A análise ao redimensionamento automático da aplicação de acordo com o tamanho do ecrã e com a orientação do dispositivo, foi efetuada através de uma ferramenta presente no *browser Google Chrome*. Esta ferramenta de análise de páginas de internet, permite aceder a uma página de internet escolhida e escolher diversos

dispositivos com tamanhos de ecrã diferentes. Do mesmo modo, é possível também mudar a orientação do dispositivo e analisar como a página de internet é adaptada a essa mudança.

Como termo de comparação, vai ser usada para análise a página onde são visualizados os artigos. Esta página é representada na figura 5.2, sendo que, esta imagem representa a página acedida através de um computador de secretária. Verifica-se a presença de uma barra de pesquisa e a existência de todas as colunas associadas a cada artigo.



Código	Designação	IVA	PV	Stock	Un.
10.001	FRT. ABOBORA BRANCA FATIADA CX 10 KG	23	2.48	580	KG
10.002	FRT. ABOBORA ENCARNADA FATIADA CX 10 KG	23	2.55	1050	KG
10.003	FRT. ABOBORA VERDE FATIADA CX 10 KG	23	2.55	1040	KG
10.004	FRT. CASCA DE LARANJA CX 10 KG	23	3.13	430	KG
10.005	FRT. CEREJA 1° CX 10 KG	23	4.8	538	KG
10.006	FRT. CHILA ESCORRIDA CX 10 KG	23	2.78	440	KG
10.008	FRT. PERA CX10 KG	23	2.7	510	KG
10.009	FRT. FIGO CX 10 KG	23	3.13	1700	KG
10.010	FRT. MARACUJA CX 10 KG	23	9.08	26	KG
10.011	FRT. CIDRAO CX10 KG	23	3.65	0	KG
10.012	FRT. TANGERINA CX 10 KG	23	2.48	326	KG
10.013	FRT. DOCE DE CHILA BL 15 KG	23	2.63	780	KG
10.015	ANANAS 3 KG	23	5.5	271	LT

Figura 5.2: Página “Artigos” acedida de um computador.

Quando acedida através de um *tablet* de dez polegadas, a página apresenta-se com menos colunas associadas a cada artigo, devido ao redimensionamento da página. No entanto, todas as colunas estão acessíveis para consulta através do botão com o nome “Colunas”. Na figura 5.3a visualiza-se o redimensionamento da página, o botão “Colunas” quando escolhido está ilustrado na figura 5.3b.

Como consequência da diminuição do ecrã, vão aparecendo cada vez menos colunas para visualizar. Pode-se constatar esse facto através da figura 5.4.

Aqui é apenas referida a página “Artigos” como título de exemplo, no entanto, todas as páginas da aplicação, cumprem com o requisito de se redimensionarem automaticamente de acordo com o tamanho do dispositivo utilizado.

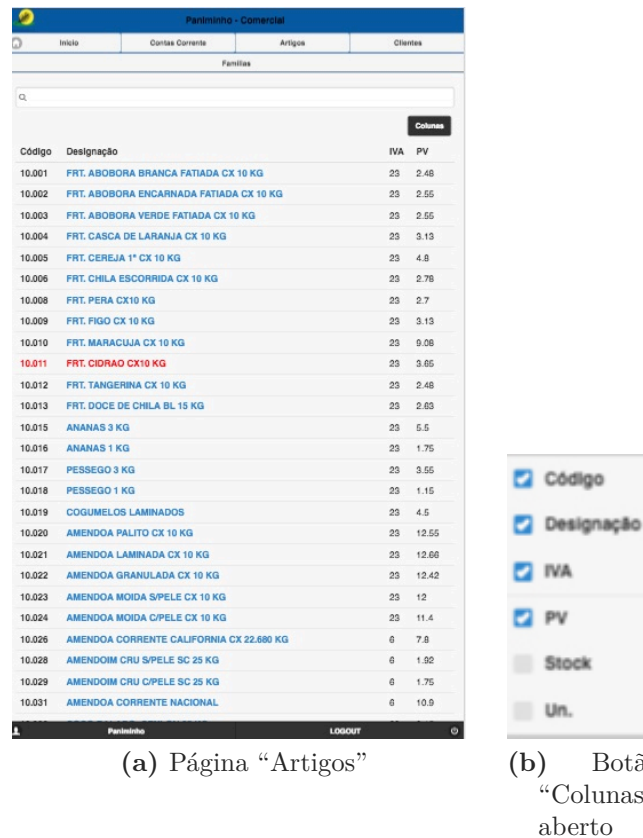


Figura 5.3: Página “Artigos” acedida de um *tablet* de dez polegadas.

5.2.2 Segurança

O processo de encriptação das *passwords* já explicado anteriormente segue um determinado procedimento. Inicialmente é criada a chave de codificação e decodificação da *password* descrita na Listagem 5.1.

Listagem 5.1: Criação da chave de segurança.

```
<?php
if(function_exists('openssl_random_pseudo_bytes')){
    echo base64_encode(openssl_random_pseudo_bytes(128));
}else{
    echo base64_encode(mcrypt_create_iv(128, MCRYPT_DEV_URANDOM));
}
```

De seguida é usada a função de *hash* para encriptação da *password* ilustrado na Listagem 5.2. É relevante referir que a função utiliza a *password* inserida e a chave

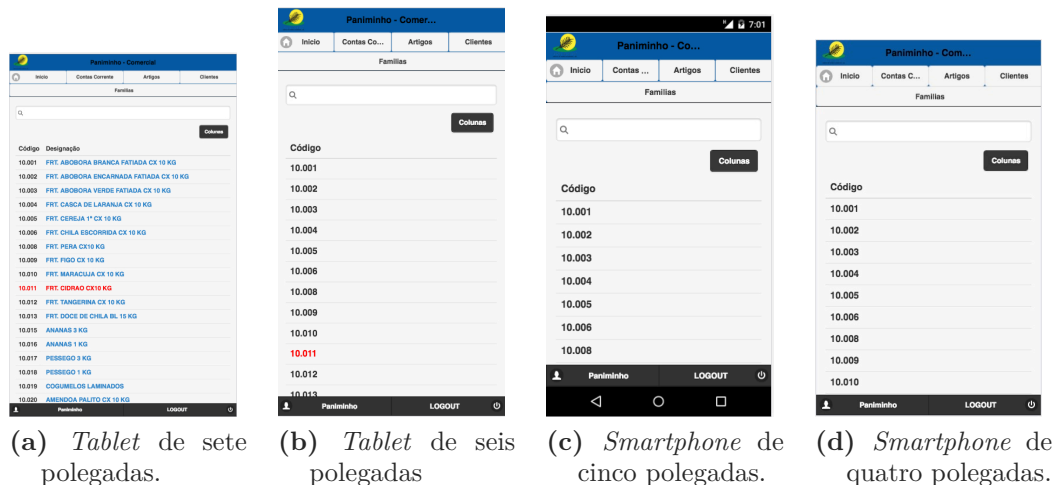


Figura 5.4: Redimensionamento da página “Artigos” em diversos dispositivos.

criada anteriormente.

Listagem 5.2: Função de *Hash*.

```
$hmac=hash_hmac( 'sha512' , $pass1 , file_get_contents( 'key.txt' ) );
```

A criação do *salt* é o seguinte passo, ilustrado na Listagem 5.3. Cada *password* possui um *salt* diferente.

Listagem 5.3: Criação do *Salt*.

```
$bytes = mcrypt_create_iv(16, MCRYPT_DEV_URANDOM);
$salt = strtr( base64_encode($bytes), '+', '.' );
$salt = substr($salt, 0, 22);
```

Por fim, é usado o algoritmo *blowfish* para criar a *password* encriptada que vai ser guardada na base de dados, como se pode visualizar na Listagem 5.4. Consegue-se visualizar que este algoritmo usa a *password* encriptada inicialmente com recurso à chave criada e ao *salt*.

Listagem 5.4: Algoritmo *Blowfish*.

```
$bcrypt = crypt($hmac, '$2y$12$' . $salt );
```

De seguida, vai ser analisada a vulnerabilidade da aplicação web com recurso a um *software* específico. De acordo com a análise efetuada, das dez principais vulnerabilidades, apenas foram encontrados problemas de segurança em três. No entanto, estes problemas encontrados estão relacionados com o servidor web onde a aplicação está alojada, bem como, com as configurações do serviço de *email*. A tabela 5.1

seguinte, ilustra as vulnerabilidades analisadas e em quais ocorreram problemas. É um resultado satisfatório e indica que a aplicação possui um nível de segurança bom.

Vulnerabilidade	
<i>Injection</i>	✓
<i>Broken Authentication and Session Management</i>	✓
<i>Cross-Site Scripting (XSS)</i>	✓
<i>Insecure Direct Object References</i>	✓
<i>Security Misconfiguration</i>	x
<i>Sensitive Data Exposure</i>	✓
<i>Missing Function Level Access Control</i>	x
<i>Cross-Site Request Forgery (CSRF)</i>	x
<i>Using Components with Known Vulnerabilities</i>	✓
<i>Unvalidated Redirects and Forwards</i>	✓

Tabela 5.1: Vulnerabilidades web.

5.2.3 Funcionalidades implementadas na aplicação web

De acordo com os objetivos pretendidos, nesta secção vai ser ilustrado através de imagens, as funcionalidades e requisitos da aplicação web.

O primeiro requisito pretendido, era a implementação de acesso autorizado à aplicação. Na figura 5.5 é visualizada a página de *login* da aplicação num dispositivo de cinco polegadas. No entanto, para que seja possível aceder à aplicação, é necessário um registo previamente efetuado pelo administrador do sistema. A página de registo é visualizada através da figura 5.6. Nesta página são inseridos dados correspondentes ao novo utilizador. Sendo eles os seguintes:

- ID.
- Nome.
- Utilizador.
- *Email*.
- Confirmação do *email*.
- *Password*.

- Confirmar *password*.
- Perfil.

De realçar o facto, de, ser necessário inserir duas vezes o *email* e duas vezes a *password*, para confirmar que estes dados foram inseridos corretamente. O campo “Nome” corresponde ao nome da pessoa, enquanto que o campo “Utilizador” é inserido para atribuir um nome pelo qual esse utilizador vai ser reconhecido na aplicação. O campo “ID” é essencial para atribuir a cada utilizador apenas os clientes que lhe pertencem, sendo que esse “ID” é também atribuído aos clientes correspondentes. Por fim, o campo “Perfil” é usado para permitir ou bloquear o acesso a determinadas páginas da aplicação *web*. Ao administrador do sistema é atribuído o perfil um, aos restantes utilizadores do sistema é lhes atribuído o perfil dois. Esta atribuição de perfis, confere apenas ao administrador do sistema o acesso total a todas as páginas da aplicação. Por outro lado, os restantes utilizadores têm acesso restrito à aplicação, sendo que não lhes é permitido o acesso a determinadas páginas.

Cada utilizador do sistema tem ao seu dispor a opção de mudar a *password* da sua conta. Esta opção é relevante, uma vez que é essencial que as *passwords* sejam alteradas de modo a garantir uma maior segurança na aplicação. A figura 5.7 ilustra a página de internet com essa função. É pedido ao utilizador que insira a *password* atual e de seguida que insira duas vezes a nova *password*.

Outro objetivo estabelecido, foi o de visualizar a conta corrente de um determinado cliente escolhido. Para além de visualizar os dados relativos à conta corrente do cliente, era também pretendido que esses dados pudessem ser guardados em ficheiro PDF e/ou guardados em ficheiro PDF e enviados por *email*. A figura 5.9 é a ilustração da página de internet que permite ao utilizador realizar essas tarefas. A página, dispõe de um botão para criar o documento PDF, como se pode visualizar, contém também um campo onde é inserido o *email* do destinatário. De realçar que quando o cliente escolhido tem um *email* associado, esse *email* aparece automaticamente. A figura 5.10 é um exemplo de um documento PDF criado, nele estão presentes os dados do cliente escolhido relativamente à sua conta corrente. É importante referir, que em cada conta corrente, caso existam documentos com atraso de pagamento de mais de noventa dias, esses mesmos documentos aparecem listados a vermelho, de modo a alertar o utilizador.

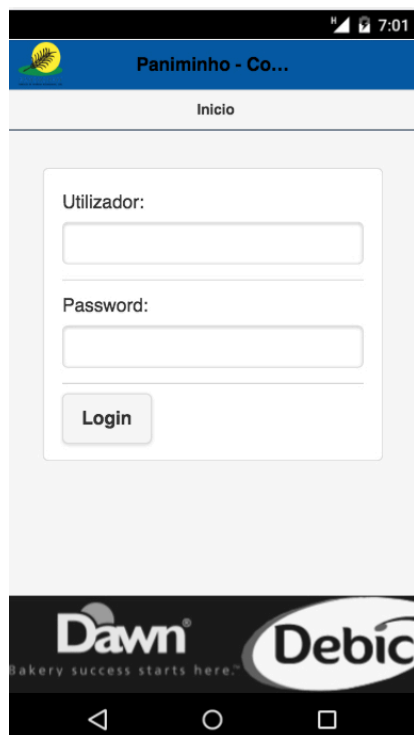


Figura 5.5: Página “Login” da aplicação *web*.

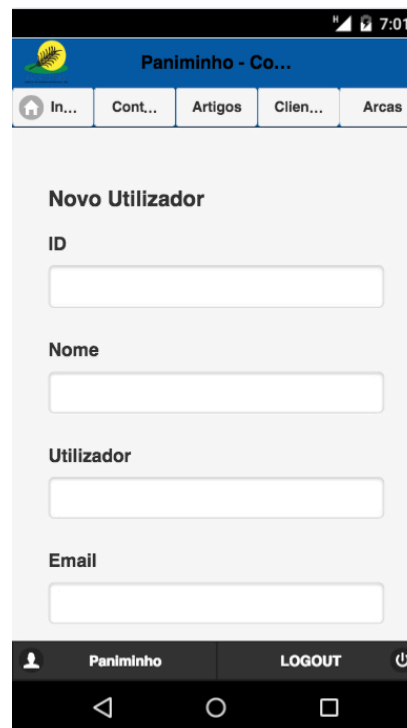


Figura 5.6: Página “Registar”.

Ainda relacionado com as contas correntes dos clientes, foi também implementada uma página designada de “Todas as Contas Correntes”, esta página foi criada com a necessidade de efetuar pesquisa por valor em todos os documentos presentes na base de dados. Esta página veio resolver um problema existente, uma vez que, muitas vezes são transferidos valores para a conta da empresa sem qualquer designação do destinatário ou do documento a liquidar. Acedendo a esta página, consegue-se pesquisar por esse valor e encontrar a que documento se refere. Na figura 5.8 está ilustrada a página mencionada onde foi pesquisado um determinado valor.

As páginas “Artigos” e “Famílias” foram criadas com o intuito de possibilitar aos utilizadores do sistema consultar os artigos disponíveis. A cada artigo é associado as suas informações essenciais. De realçar que é possível saber o *stock* existente de cada artigo. Estas funcionalidades foram também mencionadas como objetivos no primeiro capítulo deste documento. A página “Famílias” é ilustrada na figura 5.11. É possível verificar que a cada botão está associado a uma família de produtos. No entanto, quando cada família é acedida, a página não é recarregada. Isto é, quando

Username
Paniminho

Password Atual

Password Nova

Confirmar Password Nova

Atualizar

Paniminho LOGOUT

Figura 5.7: Página “Utilizador” acedida num dispositivo de quatro polegadas.

a página é acedida, todos os produtos são carregados na página, no entanto, ficam divididos por famílias. Verifica-se também, uma barra de pesquisa que auxilia a procura de um determinado artigo. A página “Artigos” já foi ilustrada anteriormente no início deste capítulo. A única diferença entre a página “Artigos” e a página “Famílias”, é o facto de na primeira, os artigos não estarem organizados por famílias. Permitindo uma pesquisa quando não se sabe a que família pertence cada artigo.

Do mesmo modo que na página “Famílias”, a página “Informação do Cliente”, permite a visualização de toda a informação relativa a cada cliente. Destaca-se o facto desta informação estar toda presente na mesma página. É possível ao utilizador consultar a morada, conta corrente e faturas do cliente. Esta página está ilustrada na figura 5.12, onde se visualizam quatro botões que depois de acedidos, possibilitam a consulta de determinada informação. É de referir também, que é possível visualizar com detalhe cada fatura do cliente em questão.

Uma das principais funções pretendidas era o acesso aos dados da temperatura. Os mesmos são ilustrados em gráficos diários, sendo que o utilizador escolhe o dia a que quer aceder. A seguinte figura 5.13 ilustra a funcionalidade implementada, que tem como principal objetivo a monitorização dos valores da temperatura de uma forma automática. Evitando a necessidade de uma deslocação às arcas frigoríficas para verificação dos valores, uma vez que um *email* de alerta é enviado para o administrador, quando os valores estão fora da gama predefinida. É de salientar, que os valores apresentados no gráfico, foram obtidos ainda em fase de testes, ou



Designação	Nº Documento	Saldo
Factura a Cliente	74979	748.13

Figura 5.8: Página “Todas as Contas Correntes” consultada num *tablet* de sete polegadas.

seja, o sistema ainda não estava instalado no local pretendido. Na figura destaca-se o calendário onde é feita a escolha do dia a que se pretende visualizar os gráficos. Este calendário aparece quando a barra de pesquisa é acedida.

Para finalizar, é relevante também referir as funcionalidades presentes na página “Atualizar”. Esta página ilustrada na figura 5.14 é a responsável pela atualização dos valores presentes na base de dados do servidor *web*. Nesta página são carregados ficheiros do tipo CSV, previamente guardados através do sistema de faturação da empresa. É importante referir também, que quando é colocado um visto em “Verificar atraso”, é enviado automaticamente para cada utilizador, um *email* com um documento PDF. Esse documento PDF contém uma listagem com os clientes atribuídos a cada vendedor e com o número de dias de atraso de pagamento que cada cliente tem. Esta funcionalidade é importante, uma vez que é um alerta para os utilizadores para terem a conta corrente de cada cliente controlada.



Figura 5.9: Página “Conta Corrente do Cliente”.



Figura 5.10: Documento PDF criado.

Concluindo, é importante referir, que todas as funcionalidades foram mais facilmente implementadas com recurso das linguagens de programação utilizadas. Mais propriamente HTML5, CSS3 e a biblioteca de funções *jQuery Mobile*. Nelas, estão presentes características que possibilitam o redimensionamento automático das páginas de internet, bem como a criação de uma *interface* com o utilizador mais simples e adaptado a qualquer dispositivo. Permitindo que a aplicação possa ser acedida de um qualquer dispositivo com ligação à internet, com diferentes dimensões de ecrã, com sistema operativo e *browser* diferentes.

5.3 Gráficos de Temperatura

Na presente secção vão ser analisados os gráficos de temperatura, onde constam os valores recolhidos pelos sensores nas respetivas arcas frigoríficas.

Tal como foi referido anteriormente, a aplicação *web* permite a visualização dos dados recolhidos pelos sensores, recorrendo a gráficos que representam os dados adqui-



Figura 5.11: Página “Famílias” acessada num *tablet* de sete polegadas.

ridos num determinado dia. Como título de exemplo foi escolhido o dia 28/10/2016, uma sexta-feira, onde estão ilustrados valores para um dia normal de trabalho.

A figura 5.15 representa os valores adquiridos na arca um. Esta arca têm como valores predefinidos de temperatura entre 1 ° e 4 ° C. Na figura, verifica-se que normalmente os valores oscilam entre os 2,1 ° e os 5,3 °C, sendo o valor máximo para esta arca 4 °C. No entanto o limite é excedido por um curto período de tempo, não sendo relevante para uma alteração das propriedades dos produtos armazenados neste local. Esta oscilação dos valores deve-se às perdas de temperatura da própria arca devido aos seus materiais de construção e isolamento, sendo que, os motores nela instalados, começam a trabalhar quando a arca atinge os 5,3 °C, reduzindo este valor para 2,1 °C. Quando este valor é atingido, os motores são desligados e a temperatura sobe novamente. Destaca-se também no gráfico, valores de temperatura de 8,4 °C, estes valores foram atingidos em momentos em que a arca frigorífica foi acessada para retirar ou repor produtos. Esta elevação dos valores deve-se ao facto de a arca possuir uma porta de entrada muito grande, logo, a temperatura vai subir rapidamente quando a porta for aberta. No entanto, destaca-se novamente, que estes valores elevados nunca são mantidos durante longos períodos de tempo, nunca pondo em causa os produtos alimentares armazenados neste local.

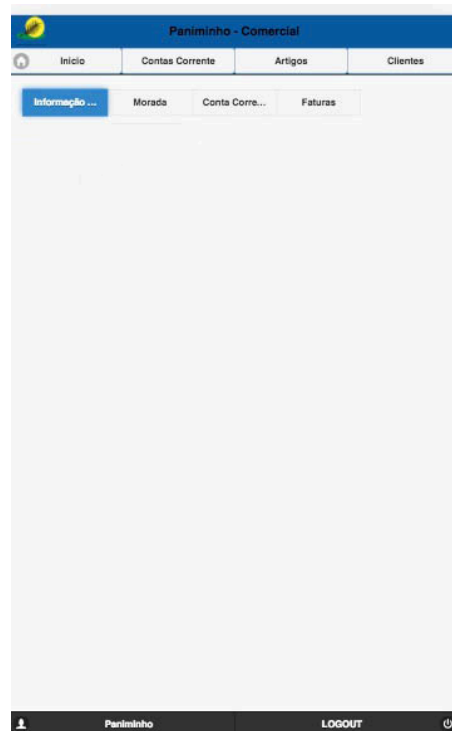


Figura 5.12: Página “Informação do Cliente” acedida num *tablet* de sete polegadas.

A figura 5.16 representa os valores obtidos na arca dois, esta arca tem uma gama de valores para a temperatura que pode oscilar entre os 4 ° e os 20 °C. No gráfico estão ilustrados dois valores distintos para a temperatura, são eles 12,6 ° e 17,4 °C. O valor mais baixo é obtido quando a arca está com as portas fechadas, no entanto, e devido ao facto da temperatura ambiente não ser muito elevada nesta altura do ano, a porta da arca frigorífica é mantida muitas vezes aberta, fazendo com que a temperatura se mantenha nos 17,4 ° C. Sendo que, a porta é mantida aberta apenas para facilitar os processos de trabalho dentro do armazém. Do mesmo modo que a arca um sofre oscilação de temperatura devido às perdas de temperatura através do seu material de construção e isolamento, também na arca dois acontece esse fenómeno. É possível verificar no gráfico essa oscilação, onde a temperatura com a porta fechada oscila entre os 9,5 ° e os 12,6 °C, e a temperatura com a porta aberta oscila entre os 12,6 ° e os 17,4 °C.

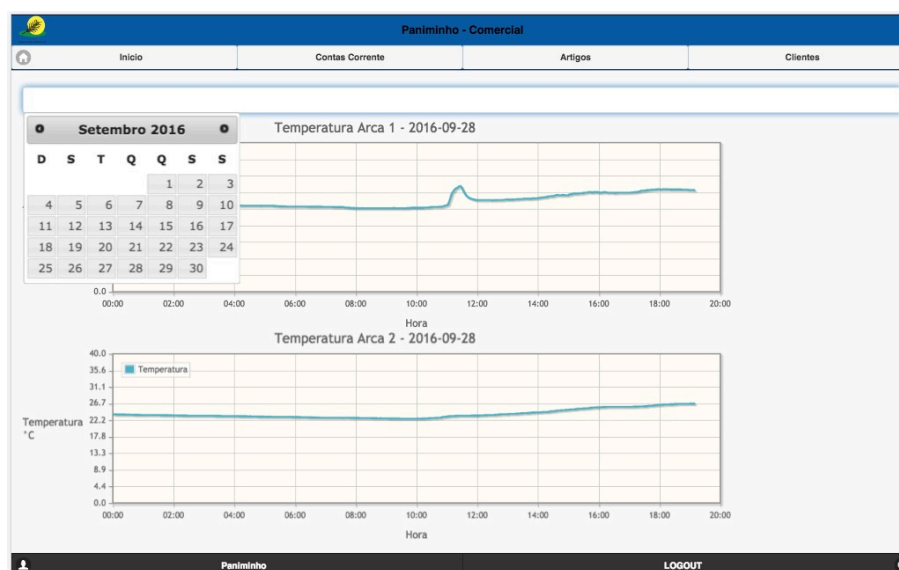


Figura 5.13: Gráficos de temperatura.

The screenshot displays a web application interface for 'Paniminho - Comercial'. At the top, there is a navigation bar with links: 'Inicio', 'Contas Corrente', 'Artigos', and 'Clientes'. The main content area contains four sections, each with a button labeled 'Escolher ficheiro' and a text label 'Nenhum ficheiro selecionado'. The sections are: 'Ficheiro Clientes:', 'Ficheiro Pendentes:', 'Artigos:', and 'Faturas:'. Below these sections are two buttons: 'Verificar atraso' and 'Atualizar'. At the bottom of the interface, there is a footer with 'Paniminho' and 'LOGOUT' buttons.

Figura 5.14: Página “Atualizar” acedida num *tablet* de sete polegadas.

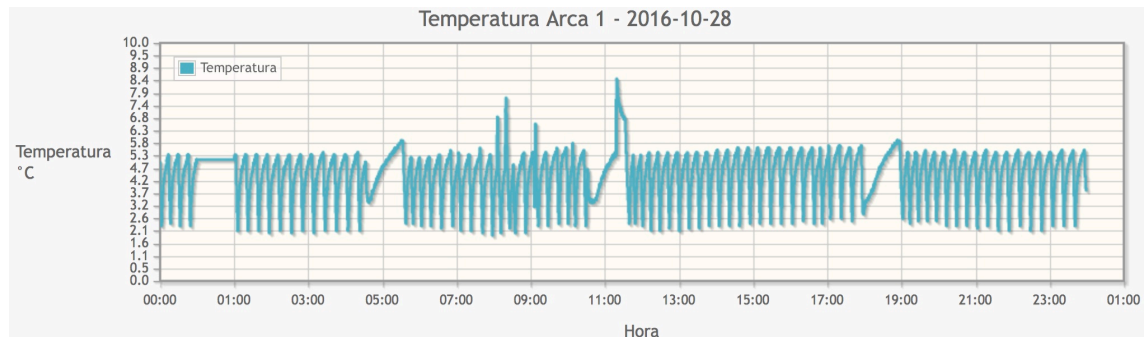


Figura 5.15: Gráfico da arca 1.

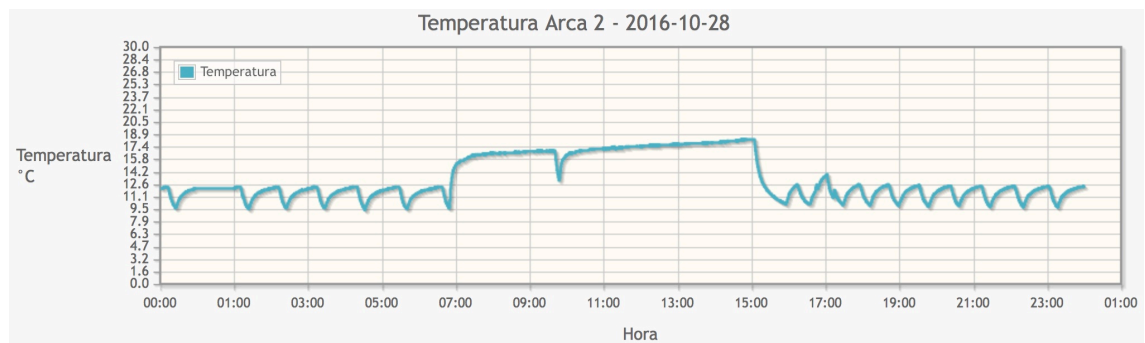


Figura 5.16: Gráfico da arca 2.

6 Conclusão e Trabalho Futuro

No presente documento académico, foi desenvolvido um projeto multidisciplinar que envolve áreas muito distintas, muitas delas desconhecidas, que levou a que tivesse sido também um processo de aprendizagem.

Relativamente ao *hardware* utilizado, a *Raspberry Pi* era desconhecida, bem como a linguagem de programação *Phyton*.

Do mesmo modo, a programação *web* era uma área desconhecida, o que levou à aprendizagem de novas linguagens de programação, sendo que, foi optado não desenvolver a aplicação *web* com recurso a uma *framework*, de modo a que fossem percebidas as bases em que assenta a programação *web*.

Em relação aos objetivos traçados, pode-se afirmar que todos eles foram cumpridos. Os dados relativos aos sensores de temperatura são recolhidos automaticamente e visualizados com recurso a uma aplicação *web*.

A aplicação *web* é também uma importante ferramenta de trabalho para a empresa em questão. Uma vez que a partir do momento que ficou a funcionar tem sido uma ajuda importante e ajudou a simplificar os processos da empresa. Mais concretamente, fornece aos utilizadores do sistema uma autonomia que não existia. Todos os dados relativamente aos artigos, clientes e conta corrente dos clientes eram fornecidos por *email* de acordo com os seus pedidos. Não só aumentou a autonomia dos utilizadores, como retirou tarefas a uma pessoa que tinha de estar dedicada a essa função, libertando-a para realizar outras tarefas.

Por outro lado, permite também um controlo maior sobre as contas correntes dos clientes, alertando os utilizadores e o administrador para clientes que se atrasem nos pagamentos.

Relativamente ao trabalho futuro, a aplicação *web* ainda precisa de melhoramentos a nível da rapidez de funcionamento, uma vez que, existem páginas *web* que demoram um pouco mais a aceder. No entanto, agora existe uma plataforma a partir da qual, se podem desenvolver mais ferramentas de trabalho, no âmbito de cada vez mais melhorar os processos da empresa. Uma ferramenta a desenvolver

brevemente, terá como objetivo simplificar a gestão das encomendas enviadas pelos utilizadores, uma vez que atualmente, estas são enviadas por *email*.

Concluindo, destaca-se o facto desta dissertação ter sido um processo de aprendizagem em que os objetivos foram cumpridos e onde foi criada uma solução que melhorou o funcionamento da empresa.

Bibliografia

- [1] V. Vujović and M. Maksimović, “Raspberry Pi as a Wireless Sensor Node : Performances and Constraints,” *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pp. 1247–1252, 2014.
- [2] I. Begum, “Remote Monitoring & Control System Using Embedded Web Server on Arm 11,” vol. 1, no. November, pp. 78–84, 2014.
- [3] M. K. D, M. S. B, and R. K. S, “Analysis of TOI (Things of Internet) Industrial Monitoring System on Raspberry pi Platform,” vol. 2, pp. 33–40, 2014.
- [4] S. Bhuvaneswari and S. A. N. A, “Implementation of Tcp / Ip on Embedded Webserver Using Raspberry Pi In Industrial Application,” vol. 3, no. 3, pp. 5240–5244, 2014.
- [5] K. L. S. H. P, D. D. G. L. Dahanayaka, and A. N. G, “Development of inexpensive multi-parameter sensors based network system for water environment monitoring,” vol. 5, no. 2, pp. 341–352, 2014.
- [6] A. A. Siddiqui, S. Amir, and N. Ahmed, “Web-based Online Parameters monitoring and control system implemented on Raspberry Pi,” 2014.
- [7] M. Eroglu S., Toprak S., Urgan O, MD, Ozge E. Onur, MD, Arzu Denizbasi, MD, Haldun Akoglu, MD, Cigdem Ozpolat, MD, Ebru Akoglu, *PHP & MySQL - the missing manual*, 2012, vol. 33.
- [8] Efa, “File:Drawing of Raspberry Pi model B rev2.svg - Wikipedia, the free encyclopedia,” 2015. [Online]. Available: https://en.wikipedia.org/wiki/File:Drawing{}_of{}_Raspberry{}_Pi{}_model{}_B{}_rev2.svg
- [9] Adafruit, “Miniature WiFi (802.11b/g/n) Module: For Raspberry Pi and more ID: 814 - \$11.95 : Adafruit Industries, Unique & fun DIY electronics and kits.” [Online]. Available: <https://www.adafruit.com/product/814>

-
- [10] Aosong Electronics, “Dht22 Datasheet,” pp. 1–7.
- [11] ITEAD, “File:RaspberryPILCD1602Add-on.jpg - ITEAD Wiki.” [Online]. Available: <https://www.itead.cc/wiki/File:RaspberryPILCD1602Add-on.jpg>
- [12] OWASP, “Top 10 2013 - OWASP,” 2015. [Online]. Available: <https://www.owasp.org/index.php/Top{ }10{ }2013>
- [13] R. Pi, “Usage - Raspberry Pi Documentation,” *www.raspberrypi.org*. [Online]. Available: <https://www.raspberrypi.org/documentation/usage/>
- [14] D. Semiconductor Corp, “Programmable Resolution 1-Wire Digital Thermometer.”
- [15] D. Bolton and B. Maunsell, “Guidelines for food safety control in European restaurants,” *Life and Management of Living Resources (QoL) Nutrition and Health. Project*, vol. 1, no. July, p. 20, 2004. [Online]. Available: <http://www3.uma.pt/jcmarques/docs/haccp/EUGuidefoodsafety.pdf>
- [16] B. Goradiya and H. Pandya, “Real time Monitoring & Data logging System using ARM architecture of Raspberry pi & Ardiuno UNO,” *Ijves.Com*, vol. 04, no. July, pp. 513–517, 2013. [Online]. Available: <http://ijves.com/wp-content/uploads/2012/07/IJVES-Y13-06118.pdf>
- [17] H. Hastriyandi, K. B. Seminar, H. Sukoco, D. C. Science, and B. Engineering, “A Multi Sensor System For Temperature Monitoring In A Greenhouse Using Remote Communication,” *International Journal of Latest Research in Science and Technology*, vol. 3, no. 4, pp. 81–87, 2014.
- [18] T. Final, G. S. C. A. Elaborado, H. Gon, F. Aluno, and N. Barcarena, “Universidade Atlântica Redes de sensores aplicadas à realidade HACCP das empresas de eventos para o controlo de temperaturas,” 2014.
- [19] M. Rodrigues, J. Mendes, and J. Fonseca, “Application of a Web-based Monitoring and Control system in Plastic Rotational Moulding Machine.”
- [20] S. Goodwin, *Smart Home Automation with Linux and Raspberry Pi*.
- [21] U. Sowmiya and S. M. J, “Raspberry Pi based home door security through 3g dongle,” vol. 3, no. 2, pp. 138–144, 2015.

- [22] J. a. Nanajkar and V. D. Nagrale, “Embedded Web Server Based Automation,” vol. 2, no. 9, pp. 265–267, 2013.
- [23] G. Birajdar, “Implementation of Embedded Web Server Based on ARM11 and Linux using Raspberry PI,” no. 3, pp. 64–66, 2014.
- [24] J. Kim, a. Biryukov, B. Preneel, and S. Hong, “On the security of HMAC and NMAC based HAVAL, MD4, MD5, SHA-0 and SHA-1,” *Security and Cryptography for Networks. 5th International Conference, SCN 2006. Proceedings (Lecture Notes in Computer Science Vol. 4116)*, vol. 2006, pp. 242 – 56, 2006.
- [25] E. Alecrim, *Criptografia*. InfoWester Propagando Conhecimento, 2010. [Online]. Available: <http://www.infowester.com/criptografia.php>
- [26] Ole Aass, “Two step password hashing with hmac and bcrypt.” [Online]. Available: <http://www.gotcode.org/blog.php?id=255>
- [27] E. Engineering, U. Teknologi, and M. Uitm, “A Web Based Temperature Monitoring System using Linux,” 2011.
- [28] M. Schmidt, *Raspberry Pi, A quick-start guide*, 2012.
- [29] J. Long, “W1thermsensor by timofurrer.” [Online]. Available: <https://timofurrer.github.io/w1thermsensor/>